REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
23-12-2014	Final with Conference Proceedings	29 Sept – 02 Oct 2014

4. TITLE AND SUBTITLE

PHM 2014 Invited Session on Corrosion Monitoring, Sensing, Detection and Prediction

5a. CONTRACT NUMBER
N00014-14-1-0817

5b. GRANT NUMBER
N00014-14-1-0817

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Karl Reichard
Carl Byington

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY, INC. DBA PHM SOCIETY
200 CANAL VIEW BLVD
ROCHESTER, NY 14623-2852

8. PERFORMING ORGANIZATION REPORT NUMBER

001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ignacio M. Perez De Leon
ONR 332
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Prognostics and Health Management (PHM) Society hosts an annual international conference which brings together the global community of PHM experts from industry, academia, and government in diverse application areas such as energy, aerospace, transportation, automotive, and industrial automation. The 2015 Conference featured a specific invited session on Corrosion Monitoring, Detection, and Prediction sponsored by ONR. A major differentiator for the PHM Society is its modern approach toward intellectual property rights. Authors retain copyright in their work while allowing the PHM Society to distribute their work through a Creative Commons License. As a result, all articles published by the PHM Society are available to the global PHM community via the internet for free and without any restrictions. A copy of the proceedings from the 2014 Conference is included as a deliverable under this grant.

15. SUBJECT TERMS
PHM, diagnostics, prognostics, health management

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Carl Byington
Unclassified	Unclassified	Unclassified	UL	1	19b. TELEPHONE NUMBER *(include area code)* (585)697-4087

www.phmsociety.org

ISBN – 978-1-936263-17-2

PHM 2014

Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014

Fort Worth, TX, USA
September 29 – October 02, 2014

ISBN – 978-1-936263-17-2

Edited by:
Matthew J. Daigle
Anibal Bregon

phmsociety

Table of Contents

Full Papers

2 Estimation of Remaining Useful Life Based on Switching Kalman Filter Neural Network Ensemble
Pin Lim, Chi Keong Goh, Kay Chen Tan, and Partha Dutta

10 A Thermodynamic Entropy Based Approach for Prognosis and Health Management
Anahita Imanian and Mohammad Modarres

20 A Low Frequency Uni-variate Model for the Effective Diagnosis and Prognosis of Bearing Signals Based Upon High Frequency Data
Jamie L. Godwin and Peter Matthews

29 Health Management and Diagnostics for Synthetic Aperture Radar (SAR) Payloads
Gregory Bower, Jonathan Zook, and Ross Bird

41 Learning Diagnosis Based on Evolving Fuzzy Finite State Automaton
Moussa Traore, Eric Châtelet, Eddie Soulier, and Hossam A. Gabbar

51 Health Monitoring of Hybrid Systems Using Hybrid Particle Petri Nets
Quentin Gaudel, Elodie Chanthery, and Pauline Ribot

62 Towards Performance Prognostics of a Launch Valve
Glenn Shevach, Mark Blair, James Hing, Larry Venetsky, Everard Martin, and John Wheelock

70 Prognostic Modeling of Valve Degradation within Power Stations
M. J. McGhee, Grant S. Galloway, Victoria M. Catterson, B. Brown, and E. Harrison

76 Validation of Model-Based Prognostics for Pneumatic Valves in a Demonstration Testbed
Chetan S. Kulkarni, Matthew Daigle, George Gorospe, and Kai Goebel

86 Remaining Useful Life Prediction of Rolling Element Bearings Based On Health State Assessment
Zhiliang Liu, Ming J. Zuo, and Longlong Zhang

92 Data-Driven Lead-Acid Battery Prognostics Using Random Survival Forests
Erik Frisk, Mattias Krysander, and Emil Larsson

102 Fatigue Crack Detection at Gearbox Spline Component using Acoustic Emission Method
Didem Ozevin, Justin Cox, William Hardman, Seth Kessler, and Alan Timmons

109 A Data Driven Method for Model Based Diagnostics and Prognostics
Michael D. Bryant

118 Planetary Gearbox Fault Diagnosis Using a Single Piezoelectric Strain Sensor
Jae Yoon, David He, Brandon Van Hecke, Thomas J. Nostrand, Junda Zhu, and Eric Bechhoefer

128 Integrated Diagnostics and Prognostics for the Electrical Power System of a Planetary Rover
Matthew Daigle, Indranil Roychoudhury, and Anibal Bregon

142 Demonstration of Prognostics-Enabled Decision Making Algorithms on a Hardware Mobile Robot Test Platform
Adam Sweet, George Gorospe, Matthew Daigle, José R. Celaya, Edward Balaban, Indranil Roychoudhury, and Sriram Narasimhan

151 Gear Diagnostics – Fault Type Characteristics
Alexander Bliznyuk, Ido Dadon, Renata Klein, and Jacob Bortman

161 Characterization of Fault Size in Bearings
Matan Mendelovich, Yitschak Sanders, Gideon Kogan, Mor Battat, Renata Klein, and Jacob Bortman

167 A Vibration-Based Approach for Stator Winding Fault Diagnosis of Induction Motors: Application of Envelope Analysis
Chao Jin, Agusmian P. Ompusunggu, Zongchang Liu, Hossein D. Ardakani, Fredrik Petré, and Jay Lee

176 Formal Verification of Complex Systems based on SysML Functional Requirements
Hoda Mehrpouyan, Irem Y. Tumer, Chris Hoyle, Dimitra Giannakopoulou, and Guillaume Brat

188 A Tool Chain for the V&V of NASA Cryogenic Fuel Loading Health Management
Johann Schumann, Vanesa Gomez-Gonzalez, Nagabhushan Mahadevan, Michael Lowry, Peter Robinson, and Gabor Karsai

198 Towards Accreditation of Diagnostic Models for Improved Performance
Anuradha Kodali and Peter Robinson

207 A Model-Based Approach for Reliability Assessment in Component-Based Systems
Saideep Nannapaneni, Abhishek Dubey, Sherif Abdelwahed, Sankaran Mahadevan, and Sandeep Neema

217 Sensor Fault Diagnosis in Quadrotors Using Nonlinear Adaptive Estimators
Remus C Avram, Xiaodong Zhang, and Jacob Campbell

225 Using Johnson Distribution for Automatic Threshold Setting in Wind Turbine Condition Monitoring System
Kun S. Marhadi and Georgios Alexandros Skrimpas

238 Temporal Causal Diagrams for Diagnosing Failures in Cyber-Physical Systems
Nagabhushan Mahadevan, Abhishek Dubey, Gabor Karsai, Anurag Srivastava, and Chen-Ching Liu

252 Case Study: Models for Detecting Low Oil Pressure Anomalies on Commercial Vehicles
Howard E. Bussey, Nenad G. Nenadic, Paul A. Ardis, and Michael G. Thurston

265 Decentralized Approach for Fault Diagnosis of Three Cell Converters
Hanane Louajri and Moamar Sayed-Mouchaweh

278 Improved Probabilistic Remaining Useful Life Estimation in Structures: Modeling Multi-site Fatigue Cracking in Oil and Gas Service Structures
Abdallah Al Tamimi and Mohammad Modarres

287 Imaging and Information Processing of Pitting-Corroded Aluminum Alloy Panels with Surface Metrology Methods
Honglei Li, Margaret R. Garvan, Jiaming Li, Javier Echauz, Douglas W. Brown, and George J. Vachtsevanos

298 A Novel Linear Polarization Resistance Corrosion Sensing Methodology for Aircraft Structure
Douglas W. Brown, Richard J. Connolly, Bernard Laskowski, Margaret R. Garvan, Honglei Li, Vinod S. Agarwala, and George J. Vachtsevanos

309 On the Use of Particle Flow to Enhance the Computational Performance of Particle-Filtering-based Prognostics
Javier A. Oliva and Torsten Bertram

320 Efficient Dependency Computation for Dynamic Hybrid Bayesian Network in On-line System Health Management Applications
Chonlagarn Iamsumang, Ali Mosleh, and Mohammad Modarres

331 Prognostic Optimization of Phased Array Antenna for Self-Healing
David Allen

338 Multiple-imputation-particle-filtering scheme for Uncertainty Characterization in Battery State-of-Charge Estimation Problems with Missing Measurement Data
David E. Acuña, Marcos E. Orchard, Jorge F. Silva, and Aramis Pérez

347 Qualitative Event-Based Fault Isolation under Uncertain Observations
Matthew Daigle, Indranil Roychoudhury, and Anibal Bregon

356 Uncertainty in Steady-State Diagnostics of a Current-Pressure Transducer: How Confident are We in Diagnosing Faults?
Shankar Sankararaman, Christopher Teubert, and Kai Goebel

367 Equipment Health Monitoring with Non-Parametric Statistics for Online Early Detection and Scoring of Degradation
Maizura Mokhtar, Joseph C. Edge, and Andrew R. Mills

378 A Generalized Machine Fault Detection Method Using Unified Change Detection
Wenyi Wang, David Forrester, and Peter Frith

392 Evolving Fuzzy Classifier based on Clustering Algorithm and Drift Detection for Fault Diagnosis Applications
Maurilio Inacio, Andre Lemos, and Walmir Caminhas

402 Novel Real-Time Nondestructive Technology for Chemical and Structural Health Management of Solid Rocket Propellants
Sami Daoud, Michal J. Villeburn, Kevin D. Bailey, and Gordon Kinloch

415 Online Normalization Algorithm for Engine Turbofan Monitoring
Jérôme Lacaille and Anastasios Bellas

423 Integrated Multivariate Health Monitoring System for Helicopters Main Rotor Drives: Development and Validation with In-Service Data
Alberto Bellazzi, Giovanni Jacazio, Bruno Maino, Gueorgui Mihaylov, Franco Pellerey, and Massimo Sorli

435 An Aircraft Lifecycle Approach for the Cost-Benefit Analysis of Prognostics and Condition-based Maintenance based on Discrete-Event Simulation
Nico B. Hölzel, Thomas Schilling, and Volker Gollnick

451 A Reference Stack for PHM Architectures
Charles Crabb

465 Managing Fleet Wide Sensory Data: Lessons Learned in Dealing with Volume, Velocity, Variety, Veracity, Value and Visibility
Preston Johnson

473 Fault Diagnosis and Prognosis Based on Lebesgue Sampling
Bin Zhang and Xiaofeng Wang

484 Fault Detection and Severity Analysis of Servo Valves Using Recurrence Quantification Analysis
M. Samadani, C. A. Kitio Kwuimy, and C. Nataraj

494 Model-Based Fault Diagnosis of a Planetary Gear Using Transmission Error
Jungho Park, Jong Moon Ha, Byeng D. Youn, Sang Hyuck Leem, Joo-Ho Choi, and Nam Ho Kim

502 Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use
Brian Bole, Chetan S. Kulkarni, and Matthew Daigle

511 Application of Unscented Kalman Filter for Condition Monitoring of an Organic Rankine Cycle Turbogenerator
Leonardo Pierobon, Rune Schlanbusch, Rambabu Kandepu, and Fredrik Haglind

519 A Probabilistic Approach for Reliability and Life Prediction of Electronics in Drilling and Evaluation Tools
Amit A. Kale, Katrina Carter-Journet, Troy A. Falgout, Ludger Heuermann-Kuehn, and Derick Zurcher

533 Are Current Prognostic Performance Evaluation Practices Sufficient and Meaningful?
Shankar Sankararaman, Abhinav Saxena, and Kai Goebel

546 A Novel Human-Machine Interface Framework for Improved System Performance and Conflict Resolution
Jiaming Li and George J. Vachtsevanos

553 Analysis Quality Index - What confidence do you have in your risk analysis?
Leila Salhi, Jacek Stecki, and Chris Stecki

561 Challenges in Concrete Structures Health Monitoring
Sankaran Mahadevan, Douglas Adams, and David Kosson

568 Investigating Vibration Properties of a Planetary Gear Set with a Cracked Tooth in a Planet Gear
Xihui Liang and Ming J. Zuo

576 Standards for Prognostics and Health Management (PHM) Techniques within Manufacturing Operations
Gregory W. Vogl, Brian A. Weiss, and M. Alkan Donmez

589 A Self-Aware Machine Platform in Manufacturing Shop Floor Utilizing MTConnect Data
Linxia Liao, Raj Minhas, Arvind Rangarajan, Tolga Kurtoglu, and Johan de Kleer

597 Fault Tolerant Control for manufacturing discrete systems by filter and diagnoser interactions
A. Philippot, P. Marangé, F. Gellot, J.F. Pétin, and B. Riera

604 A Novel Feature Extraction Method for Monitoring (Vehicular) Fuel Storage System Leaks
Fling Tseng, Imad H. Makki, Dimitar P. Filev, and Ratna Babu Chinnam

612 Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset
Emmanuel Ramasso and Abhinav Saxena

623 Enhanced Trajectory Based Similarity Prediction with Uncertainty Quantification
Jack Lam, Shankar Sankararaman, and Bryan Stewart

635 Survey of Condition Indicators for Condition Monitoring Systems
Junda Zhu, Thomas J. Nostrand, Cody Spiegel, and Brogan Morton

648 Estimation of Bogie Performance Criteria Through On-Board Condition Monitoring
Parham Shahidi, Dan Maraini, Brad Hopkins, and Andrew Seidel

658 Particle Filtering-Based System Degradation Prediction Applied to Jet Engines
Peng Wang and Robert X. Gao

664 Building a Data-Driven Vital Sign Indicator for an Economically Optimized Component Replacement Policy
Hyung-il Ahn, Ying Tat Leung, and Axel Hochstein

676 Aircraft Hydraulic System Leakage Detection and Servicing Recommendations Method
Wlamir O. L. Vianna and João Pedro P. Malere

681 Dependable Wireless Sensor Networks for Prognostics and Health Management: A Survey
W. Elghazel, J.M. Bahi, C. Guyeux, M. Hakem, K. Medjaher, and N. Zerhouni

Poster Papers

693 Cost-Benefit Analysis and Specification of Component-level PHM Systems in Aircrafts
Alexander Kählert, Sebastian Giljohann, and Uwe Klingauf

705 A Framework for Model-Based Diagnostics and Prognostics of Switched-Mode Power Supplies
Honglei Li, Xuerong Ye, Cen Chen, and George J. Vachtsevanos

713 Anomaly Detection Techniques for the Condition Monitoring of Tidal Turbines
Grant S. Galloway, Victoria M. Catterson, Craig Love, and Andrew Robb

725 Detection of Wind Turbine Power Performance Abnormalities Using Eigenvalue Analysis
Georgios Alexandros Skrimpas, Christian Walsted Sweeney, Kun S. Marhadi, Bogi Bech Jensen, Nenad Mijatovic, and Joachim Holbøll

732 A PHM Approach to Additive Manufacturing Equipment Health Monitoring, Fault Diagnosis, and Quality Control
Jae Yoon, David He, and Brandon Van Hecke

741 Modeling Degradation Using Thermodynamic Entropy
Michael D. Bryant

748 Electrochemical Impedance Spectroscopy for Coating Evaluation using a Micro Sensor
Nicholas Waters, Richard J. Connolly, Douglas W. Brown, and Bernard Laskowski

754 A Time-domain Modeling and Simulation Framework for Comparative Analysis of Prognostics, Reliability and Robustness in System Design
Nicholas A. Lambert, Kyle B. Ferrio, and Douglas L. Goodman

762 Eddy Currents Signatures Classification by Using Time Series: a System Modeling Approach
Blaise Guépié, Mihaly Petreczky, and Stéphane Lecoeuche

770 Operation Condition Monitoring using Temporal Weighted Dempster-Shafer Theory
Xiaoyun Wang and Tingdi Zhao

775 Modeling Hydraulic Components for Automated FMEA of a Braking System
Peter Struss and Alessandro Fraracci

785 **Author Index**

Full Papers

Estimation of Remaining Useful Life Based on Switching Kalman Filter Neural Network Ensemble

Pin Lim[1], Chi Keong Goh[2], Kay Chen Tan[3] and Partha Dutta[4]

[1,2,4] *Rolls Royce Singapore, Singapore*
Pin.Lim@Rolls-Royce.com
ChiKeong.Goh@Rolls-Royce.com
Partha.Dutta@Rolls-Royce.com

[3] *National University of Singapore, Singapore*
eletankc@nus.edu.sg

ABSTRACT

The proposed method is an extension of an existing Kalman Filter (KF) ensemble method. While the original method has shown great promise in the earlier PHM 2008 Data Challenge, the main limitation of the KF ensemble is that it is only applicable to linear models. In prognostics, degradation of mechanical systems is typically non-linear in nature, therefore limiting the applications of KF ensemble in this area. To circumvent this problem, this paper propose to approximate non-linear functions with piecewise linear functions. When estimating the RUL, the Switching Kalman Filter (SKF) is able to choose the most probable degradation mode and thus make better predictions. The implementation of the proposed SKF ensemble method is illustrated by implementing on NASA's C-MAPSS Dataset as well as the PHM 2008 Data Challenge Dataset. The results show the effectiveness of the SKF in detecting the switching point between various degradation modes as well as the improved accuracy of the SKF ensemble method compared to other available methods in literature.

1. INTRODUCTION

In the recent years, Condition Based Maintenance (CBM) has been garnering more attention as it allows industries to better plan logistics as well as save cost by replacing parts only when needed. Prognostics being one of the key enablers of CBM has therefore also gained more interest in both academia and industry. The key notion of prognostics, albeit not the only one, is to determine the time remaining before a likely failure. This value is commonly termed as the Remaining Useful Life (RUL) of the system.

Pin Lim et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In this paper, a novel prediction algorithm is presented which is applicable to non linear degradation models. The algorithm assumes that degradation model can be described by a number of piece-wise linear functions. With each of these linear functions describing a linear model, the most suitable model to describe the degradation at any point in time is chosen based on the Switching Kalman Filter (SKF) algorithm. The remainder of this paper is structured as follows, Section 2 first introduces the datasets used to evaluate the effectiveness of the algorithm. Section 3 follows by presenting a simple single neural network approach to evaluate the difficulty of the problem. Finally in Section 4 the SKF ensemble approach is presented and evaluated.

2. DATASET

In this paper a total of two datasets were used. The datasets used are namely the PHM 2008 Data Challenge Dataset as well as the NASA C-MAPSS Dataset (Saxena & Goebel, 2008), the C-MAPSS dataset is further divided into 4 sub-datasets as shown in Table 1. Both datasets contain simulated data produced using a model based simulation program (named Commercial Modular Aero-Propulsion System Simulation, C-MAPSS) developed by NASA (Saxena, Goebel, Simon, & Eklund, 2008).

Table 1. Dataset details (Simulated from C-MAPSS)

	C-MAPSS				PHM
Dataset	**FD001**	**FD002**	**FD003**	**FD004**	**2008**
Train Trajectories	100	260	100	248	218
Test Trajectories	100	259	100	248	218
Conditions	1	6	1	6	6
Fault Modes	1	1	2	2	2

The data is arranged in an n-by-26 matrix where n corresponds to the number of data points in each dataset. Each row is a snapshot of data taken during a single operational cycle and each column represents a different variable. Included in the data are three operational settings that have a substantial effect on engine performance.

Each trajectory within the train and test trajectories is assumed to the be life-cycle of an engine. While each engine is simulated with different initial conditions, these conditions are considered to be of normal conditions (no faults). For each engine trajectory within the training sets, the last data entry corresponds to the moment the engine is declared unhealthy. On the other hand the test sets terminate at some time prior to failure and the aim is to predict the number of Remaining Useful Life (RUL) of each engine of the test set.

For each of the C-MAPSS dataset the actual RUL value of the test trajectories were made available to the public while the actual RUL of the test dataset of PHM 2008 is not available. However, users can submit their results to the NASA website to obtain a score limited to one submission per day. Due to this constrain, most of the analysis done in this paper will be based on the NASA C-MAPSS dataset instead of the PHM 2008 dataset. The PHM 2008 dataset would instead be used for comparison against other algorithms proposed in literature.

2.1. Evaluation Metrics

2.1.1. Scoring Function

The scoring function used in this paper is identical to that used in PHM 2008 Data Challenge. This scoring function is illustrated in Eq. (1), where s is the computed score, N is the number of engines, and $d = \bar{R}UL\text{-}RUL$ (Estimated RUL- True RUL).

$$s = \sum_{i=1}^{N} s_i, s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1 & \text{for } d_i < 0 \\ e^{\frac{d_i}{10}} - 1 & \text{for } d_i \geq 0 \end{cases} \quad (1)$$

The characteristic of this scoring function is that it favours early predictions more than late predictions. This is in line with the risk adverse attitude in aerospace industries. However there are several drawbacks with this function. The most significant drawback being a single outlier would dominate the overall score, thus masking the true accuracy of the algorithm. Another drawback is the lack of consideration of the prognostic horizon of the algorithm. The prognostic horizon assess the time before failure which the algorithm is able to accurately estimate the RUL value within a certain confidence level. Finally this scoring function favours algorithms which artificially lowers the score by underestimating the RUL. Despite all these shortcomings, this scoring function is still used in this paper in order to provide a level comparison with other methods in literature.

2.1.2. RMSE

In addition to the scoring function, the Root Mean Square Error (RMSE) of the estimated RULs is also used as a performance measure. RMSE is chosen as it gives equal weight to both early and late predictions. Using RMSE in conjunction with the scoring function would prevent the user from favouring an algorithm which artificially lowers the score by underestimating but resulting in higher RMSE. Furthermore, various literature working on this dataset uses RMSE to evaluate their algorithms, inclusion of RMSE would therefore allow the author to compare results with those available in literature.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \quad (2)$$

A comparative plot between the two evaluation metrics is shown in Figure 1. It can be observed that at lower absolute error values the scoring function results in lower values than the RMSE. The relative characteristics of the two evaluation metrics will be useful during the discussion of results in the latter part of this paper.

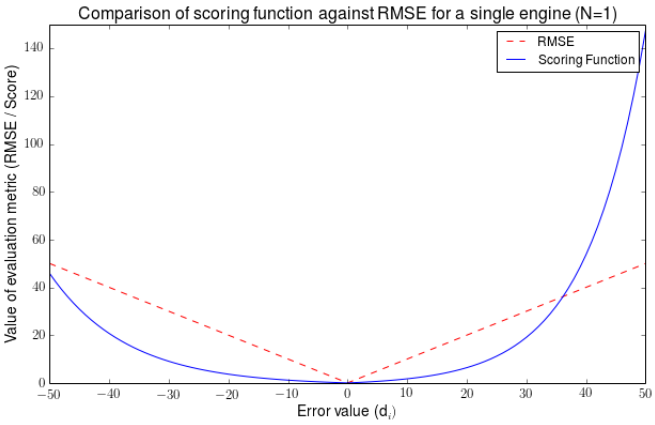


Figure 1. Comparison of evaluation metric values for different error values

2.2. Data Preparation

2.2.1. Operating Conditions

Several literature (Wang et al., 2008; Peel, 2008; Heimes, 2008), have shown that by plotting the operational setting values, the data points are clustered into six different distinct clusters. This observation is only applicable for datasets with different operational conditions, data points from FD001 and FD003 are all clustered at a single point instead. These clusters are assumed to correspond to the six different oper-

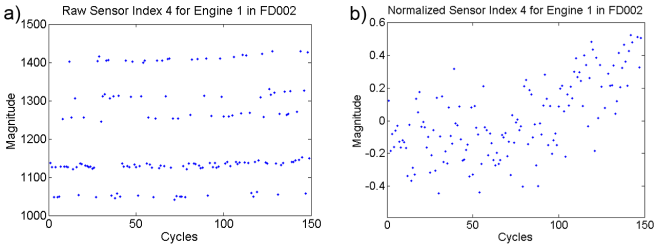


Figure 2. Sensor values (a) before and (b) after normalization

ational conditions. It is therefore possible include the operational condition history as a feature. This is done by adding 6 columns of data representing the number of cycles spent in their respective operational condition since the beginning of the series (Peel, 2008).

2.2.2. Data Normalization

Due to the 6 operating conditions, each of these operating conditions results in disparate sensor values as shown in Figure 2. Therefore prior to any testing and training, it is imperative to normalize the data points to be within the range of [-1,1] using Eq. (3). As normalization was carried out within the range of values for each sensor and each operating condition, this will ensure equal contribution from all features across all operating conditions (Peel, 2008). Alternatively, it is also possible to incorporate operating condition information within the data to take into consideration various operating conditions

$$Norm(x^{(c,f)}) = 2\frac{(x - x_{min}^{(c,f)})}{x_{max}^{(c,f)} - x_{min}^{(c,f)}} - 1, \forall c, f \qquad (3)$$

where c represents the operating conditions and f represents each of the original 21 sensors.

3. SINGLE NEURAL NETWORK APPROACH

3.1. Method Description

The aim of this section is two-fold. Firstly as a prior to experimenting with other methods, the complexity of the problem was tested using a single Multi-Layer Perceptron (MLP) Network to achieve a baseline performance. This baseline performance then used for comparing the accuracy of the proposed method. Secondly, the method is used to evaluate the performance of the two different RUL functions presented in section 3.2 below.

3.2. Arbitrary RUL Function

In its crudest form prognostic algorithms are similar to regression problems. However, unlike typical regression problems, an inherent challenge for data driven prognostic problems is determining the desired output values for each input data point. This is because in real world applications, it is impossible to accurately determine the health of the system at each time step without an accurate physics based model. A sensible solution would be to simply assign the desired output as the actual time left before functional failure (Peel, 2008; Baraldi, Mangili, & Zio, 2012). This approach however inadvertently implies that the health of the system degrades linearly with usage (Figure 3a).

An alternative approach is to derive the desired output values based on a suitable degradation model. For this data-set (Heimes, 2008) has proposed a piece-wise linear degradation model which limits the maximum value of the RUL function (Figure 3b). The maximum value was chosen based on the observations of the data and its numerical value is different for each data-set. For the sake of simplicity, the former will be addressed as 'linear function' while the latter will be known as the 'kink function' in the remainder of the paper.

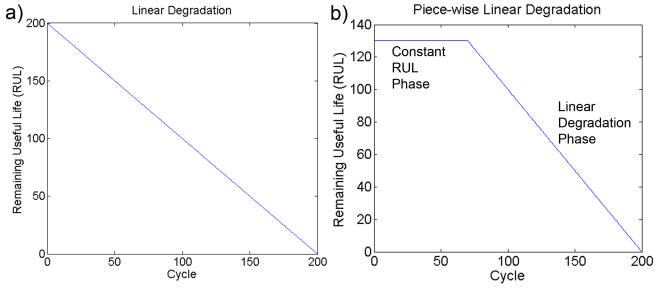


Figure 3. Comparison of degradation models. a) Linear Degradation model, b) Piece-wise Linear Degradation Model

Each of these approaches has their own advantages. The latter case is more likely to prevent the neural network from overestimating the RUL, it is also a more logical model as the degradation of the system typically only starts after a certain degree of usage. On the other hand, the former case follows the definition of RUL in the strictest sense which defined as the time to failure. Therefore the plot of time left of a system against the time passed naturally results in a the linear function as shown in Figure 3a. However it should be noted that in cases where knowledge of a suitable degradation model is unavailable, the linear model is the most natural choice to use.

3.3. Results

For each sub-dataset within the C-MAPSS dataset, two MLPs were individually trained using the linear and kink RUL functions as desired outputs. The MLPs were then tested using the corresponding test sub-datasets and evaluated using Eq. (1) and Eq. (2). Due to the inherent noise in the data, in order to capture the variance of each MLP, the whole training and testing process was repeated for a total of 10 trials. The results from these trials are expressed in the form of box plots shown in Figure 4 & 5.

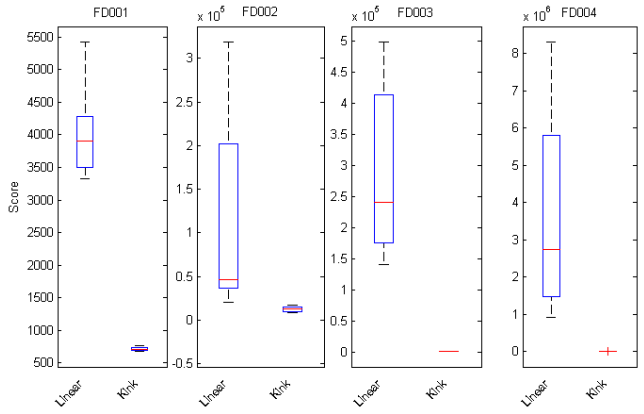


Figure 4. Scores of MLP trained with linear and kink RUL functions.

Figure 4 shows that using the linear RUL function resulted in comparatively much higher variance in scores. However considering the RMSE plots (Figure 5) the variance of RMSE values within each dataset is relatively similar. Therefore the higher variance in scores is due to the nature of the scoring function. The exponential term in the scoring function could cause large deviations in the score due to a single inaccurate estimation. The variance of the RMSE values for both MLPs could be attributed to the inability of the single MLP to handle noisy input data.

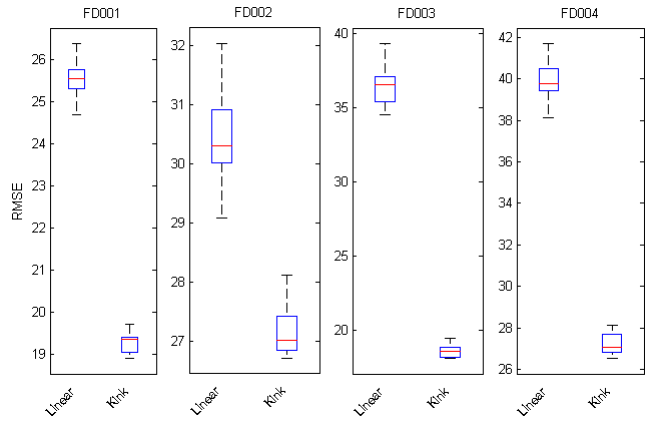


Figure 5. RMSE of MLP trained with different RUL functions.

More importantly, all datasets show significant improvements in both RMSE and scores when the kink RUL function is used. The lower RMSE values obtained by using the kink RUL function (Figure 5) is evidence that their respective lower scores in Figure 4 is due to more accurate predictions instead of inducing underestimation of RUL. These results agree with Heimes (2008) that the kink RUL function is a much more suitable degradation model for these datasets.

4. SWITCHING KALMAN FILTER (SKF) ENSEMBLE

4.1. Method Description

In order to improve the prognostic accuracy of a single MLP implemented in section 3.3, ensemble methods are explored to develop a more accurate and robust prognostic method. Ensemble methods are generally used to combine multiple weak classifiers into a single strong classifier. It has been found that ensembles would have higher accuracy and generalizability if each ensemble members are accurate and make errors on different parts of the input space (Maclin & Opitz, 2011). There are generally two main steps in creating an ensemble: The first step is to create individual ensemble members, and the second step to combine the output of the ensemble members.

In order for the ensemble to generate better results, the generalization of the ensemble must be improved. This can be obtained by having diversity in the ensemble members. The most commonly used method to create ensemble members include input data sampling techniques such as Bagging and Boosting (Zhou, 2012; Re & Valentini, 2011). In this paper, networks with different network topology are used to create ensemble members as this method has less variables to tune as compared to boosting and bagging.

Combination of output from ensemble members is usually taken as a weighted mean or median of the ensemble member outputs (Zhou, 2012). The weights are usually determined based on the training error of each ensemble member (Krogh & Vedelsby, 1995). Peel (2008) proposed an alternative combination method which uses a Kalman filter to combine the output of several neural networks. This method has shown great promise by wining the IEEE Gold for PHM 2008 Data Challenge. In his work, both the training function for the neural networks and the model used in the Kalman filter assumes a linear degradation function thus limiting its application to linear cases. This section extends this method by using a Switching Kalman Filter (SKF) for piecewise linear applications. Thus allowing implementation of a similar ensemble for other degradation patterns.

4.2. Ensemble Members

In this paper MLPs with different number of hidden neurons are used as ensemble members. The number of hidden neurons were randomly picked from a uniform distribution of integers between 5 to 25 inclusive. The maximum number of hidden neurons was limited to prevent over fitting on the training set, thus ensuring generalization on unseen data points. A total of 4 ensemble members were generated per ensemble.

4.3. Aggregation based on Kalman Filter (KF)

KFs and its variants have been widely used for machine learning applications. These applications range from simple state

prediction (Borguet & Léonard, 2009) to training of neural network weights using the Extended Kalman Filter (EKF) (Singhal & Wu, 1989; Puskorius & Feldkamp, 1991). In this paper, the traditional KF and its variant the SKF will be used.

4.3.1. Kalman Filter

The more commonly used application of the KF is as a forward pass state estimator. The filter predicts the hidden states for the next time step given the history of estimated states and observing noisy outputs. The predicted states are considered optimal as the filter aims to minimize the uncertainties in the estimate (AL-Mathami, Everson, & Fieldsend, 2012). Prior to using the KF, the system must be modeled as a linear system as shown

$$\begin{aligned} x_t &= Ax_{t-1} + w_t \\ z_t &= Hx_t + v_t \end{aligned} \tag{4}$$

where x_t is the state vector at time t, A is the transition matrix, z_t represents the output observations, H is the observation matrix, w_t and v_t are the process noise and observation noise respectively. Based on the model a recursive process is then carried out whereby the prediction step is carried out by

$$\begin{aligned} \hat{x}_t &= A\bar{x}_{t-1} \\ \hat{P}_t &= A\bar{P}_{t-1}A^T + Q \end{aligned} \tag{5}$$

where P_t is the state covariance matrix and Q is the process error covariance matrix. The KF then updates the estimate based on the new observations. The updating step is then carried out by the following equations

$$\begin{aligned} K_t &= \hat{P}_t H^T [H\hat{P}_t H^T + R]^{-1} \\ \bar{x}_t &= \hat{x}_t + K_t[z_t - H\hat{x}_t] \\ \bar{P}_t &= [I - K_t H]\hat{P}_t \end{aligned} \tag{6}$$

where R is the observation error covariance matrix and K_t is the Kalman gain at time t. For illustrative purposes, the state x_t is chosen as

$$x_t = \begin{bmatrix} RUL_t \\ \Delta RUL_t \end{bmatrix}, \Delta RUL_t = RUL_t - RUL_{t-1} \tag{7}$$

It is therefore straight forward to express the kink RUL function as a piecewise linear function with their respective linear KF model expressed as

$$A_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \; A_l = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{8}$$

where A_c is the model for the initial constant RUL phase and A_l is the model for the linear degradation phase, assuming a gradient of -1 for the linear degradation phase. In addition, the outputs from individual neural networks are taken to be the observations, therefore the observation vector z_t and H are set as

$$z_t = \begin{bmatrix} R\hat{U}L_1 \\ ... \\ R\hat{U}L_n \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ ... & ... \\ 1 & 0 \end{bmatrix} \tag{9}$$

where $R\hat{U}L_n$ is the output of the n^{th} neural network in the ensemble. Further details of modeling the ensemble outputs is covered in Peel (2008) and Baraldi et al. (2012).

4.3.2. Kalman Smoother

In contrast to the KF, which estimates the optimal state given observations up to time t, the Kalman smoother aims to estimate the optimal state at time t given the observations from 1 to T, where T represents the total length of data observations (AL-Mathami et al., 2012). The Kalman smoother is an analogous backwards recursive process which estimates the states from the end of the observation data. Therefore combining both forward and backward pass gives the optimal estimated state given the whole observation data.

At the last time step the variables $\tilde{x}$ and $\tilde{P}$ are initialized as

$$\begin{aligned} \tilde{x}_T &= \bar{x}_T \\ \tilde{P}_T &= \bar{P}_T \end{aligned} \tag{10}$$

where $\tilde{x}$ is the smoothed state and $\tilde{P}$ is the smoothed covariance. The smoothed states can then be calculated based on the following recursive equations where t decreases from $T-1$ to 1 (AL-Mathami et al., 2012).

$$\begin{aligned} J_t &= (\bar{P}_t A^\top)\hat{P}_{t+1}^{-1} \\ \tilde{x}_t &= \bar{x}_t + (J_t(\tilde{x}_{t+1}^\top - A\tilde{x}_t^\top))^\top \\ \tilde{P}_t &= \bar{P}_t + J_t(\tilde{P}_{t+1} - \hat{P}_{t+1})J_t^\top \end{aligned} \tag{11}$$

4.4. Switching Kalman Filter (SKF)

Eq. (8) in the earlier section has shown that the Kink degradation function can be modeled using two linear systems. The outputs of the ensemble members would therefore need to be combined using the suitable KF model. This problem is further compounded by the fact that the switching point between the two models differ for every engine. Thus making it difficult to pre-define a rule to switch between the two models. To circumvent this problem a SKF (Murphy, 1998; AL-Mathami et al., 2012) is implemented to autonomously determine the switching point.

In this application, SKF predicts the most probable hidden discrete model given the observations and the models. The

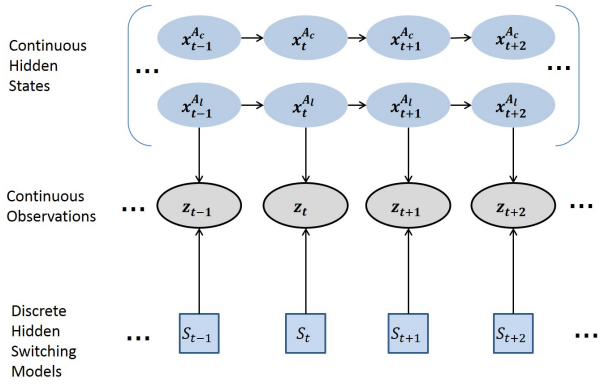


Figure 6. Directed graphical probabilistic model of SKF

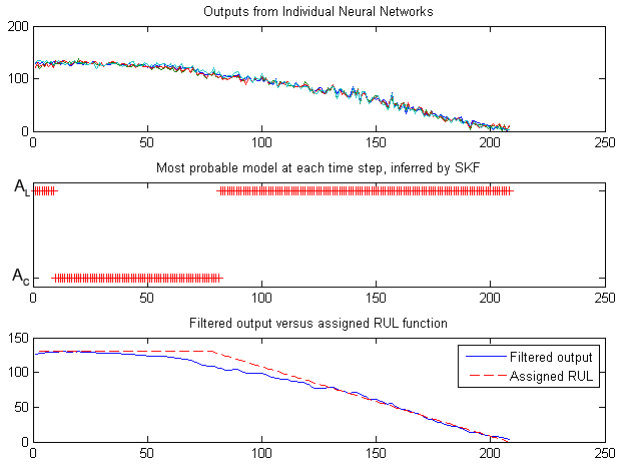


Figure 7. Example of SKF Ensemble output on a training engine

graphical probabilistic model of the SKF for aggregating ensemble methods is shown in Figure 6. Based on the figure, the SKF determines the sequence of models which would most likely result in the series of observations. Similar to the KF, the SKF computes the posterior probability of the model given the observations in two passes. The forward pass calculates $P(S_t = j|x_t, x_{1:t-1})$ while the backwards pass calculates $P(S_t = j|x_{t:T})$. An illustrative example of the forward pass calculation is shown below

For each t, j:

$$\begin{aligned} P(S_t = j|x_t, x_{1:t-1}) &= \frac{P(x_t|S_t=j, x_{1:t-1})P(S_t=j|x_{1:t-1})}{P(x_t|x_{1:t-1})} \\ &= \tfrac{1}{c} L_t(j)\Sigma_i Z(i,j)P(S_{t-1} = i|x_{1:t-1}) \end{aligned} \tag{12}$$

where

$$\begin{aligned} &c = P(x_t|x_{1:t-1}) = \Sigma_j L_t(j)\Sigma_i Z(i,j)P(S_{t-1} = i|x_{1:t-1}) \\ &L_t = P(x_t|S_t = j, x_{1:t-1}) \sim N(x_t, A_j x_{t-1}, Q_j) \\ &Z(i,j) = P(S_t = j|S_t = i, x_{1:t-1}) \end{aligned} \tag{13}$$

It should be noted that $Z(i, j)$ is a predefined transition matrix which contains the probability of transition from one model to another. Thus, based on this calculated probability, the most probable model can be chosen. The backwards pass can be calculated in a similar manner and therefore will not be repeated here. For more details on the SKF, readers can refer to Murphy (1998) and AL-Mathami et al. (2012)

In this implementation, the output of the trained ensemble members are taken to be the observations and switching models corresponds to the two KF models expressed in Eq. (8). The most probable sequence of models is first determined by the SKF, the corresponding KF models can then be applied to aggregate the outputs of individual ensemble members to obtain the estimated RUL value. Figure 7 shows an example of the SKF algorithm estimating the degradation of an engine from the training set. It can be observed that the predicted switching point between the two models by the SKF corresponds well with the predefined kink location in the RUL function. It should also be noted that the initial conditions of the Kalman filter is re-initialized for each engine.

4.5. Results

In this section the performance of the SKF ensemble is illustrated and compared with the original KF ensemble method. The KF ensemble was recreated to the best of knowledge based on the details given in Peel (2008). Furthermore, results obtained from Section 3.3 are also included for comparison purposes to highlight the effectiveness of ensemble methods. Similar to previous sections, all the experiments were repeated for a total of 10 trials, the results obtained from these trials are then expressed in the form of a boxplot.

4.5.1. C-MAPSS Dataset

Figure 8 illustrates the scores of all methods described in this paper for all four sub-datasets within C-MAPSS. It is observed that both linear MLP or KF ensemble displayed high mean and large variance of scores. In addition all four methods achieved RMSE values of the same order (Figure 9). Based on these observations, coupled with the characteristics of each evaluation metric (Figure 1), it can be implied that the high scores are caused by certain outliers in predicting the RUL. This phenomenon could probably be attributed to the use of the linear RUL function which might lead to overestimating of the RUL, thus resulting in significantly higher scores.

In addition, the high scores exhibited by the Linear MLP and KF ensemble resulted in a badly scaled boxplot making it difficult to illustrate and compare the relative performance of the remaining algorithms. Therefore more in depth comparison of the four methods will focus mainly on the RMSE values instead (Figure 9).

Based on Figure 9, it can also be deduced that the SKF ensemble outperforms that KF ensemble significantly. The SKF ensemble achieved much lower RMSE values which is most likely attributed to the use of the kink RUL function to model the degradation of the system. These results reaffirm the hypothesis arrived in Section 3.3 that the kink RUL function is a much more accurate model for this dataset.

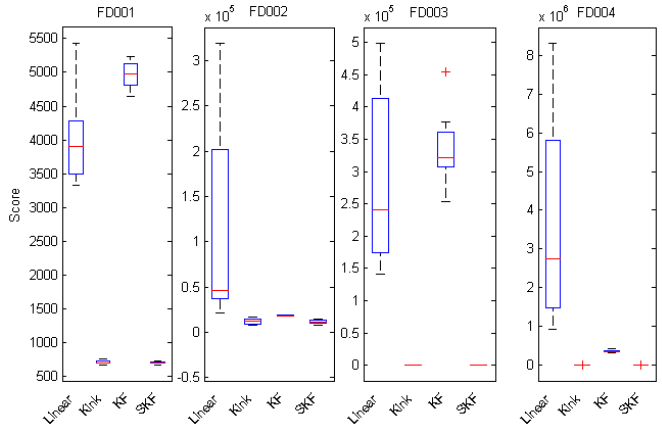


Figure 8. Scores of various algorithms for all C-MAPSS Datasets.

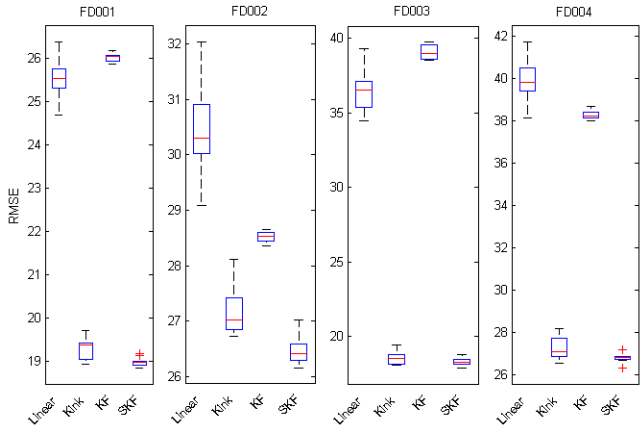


Figure 9. RMSE of various algorithms for all C-MAPSS Datasets.

As expected, both KF and SKF ensemble methods resulted in significantly lower RMSE variance compared to their respective linear and kink MLPs. This can be attributed to the ability of ensembles to aggregate the outputs of individual ensemble members thus resulting in a lower variance. In addition, the use of KF helps to filter out noise from the output of the ensemble (Figure 7) thus resulting in increased robustness against inherent noise in the data. The same observations can be seen in Figure 10 which shows in greater detail the comparison box plot between the SKF ensemble and the single MLP trained with a kink training function. In addition to obtaining lower variance in RMSE values, the SKF ensemble also exhibited lower mean RMSE values. Thus showing that the SKF ensemble outperforms the original MLP in both accuracy and variance in predictions.

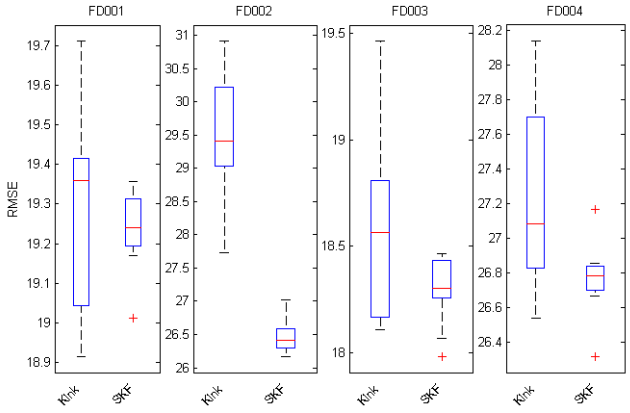


Figure 10. RMSE of MLP with Kink training function and SKF for all C-MAPSS Datasets.

Comparing the scores between the Kink MLP and the SKF ensemble (Figure 11) for all datasets showed that both methods achieved scores within the similar range. However the SKF slightly out performs the Kink MLP by exhibiting less variance in scores throughout the 10 trials. This phenomenon can be similarly be attributed the ability of ensemble to be more robust to noise as mentioned in the earlier paragraph.

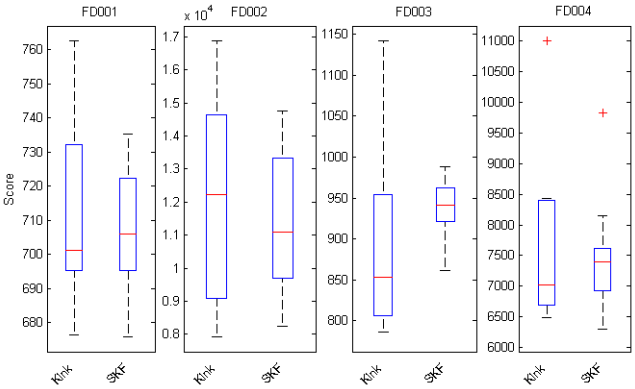


Figure 11. Scores of MLP with Kink training function and SKF for all C-MAPSS Datasets.

4.5.2. PHM 2008 Dataset

In this section, the algorithms were tested on the test dataset for PHM 2008. The estimated RULs of 218 engines within the dataset were then uploaded to the NASA Data Repository website and a single score was then returned by the website. The results were also compared with available literature that provided suitable scores for comparison.

Based on the results it can be seen that the SKF ensemble produces significantly lower scores and outperforms the other

Table 2. Scores for various algorithms on PHM 2008 test dataset

Methods	Scores
Single MLP (Linear)	118338
Single MLP (Kink)	6103.46
KF Ensemble	5590.03
SKF Ensemble	2922.33
Gibbs Filtering (Le Son, Fouladirad, & Barros, 2012)	4170

methods. However as mentioned in Section 2, submission of estimated RULs are limited to once a day. Thus the scores shown in Table 3 are from a single submission. Therefore these scores are also subject to variance as seen in earlier sections.

5. CONCLUSION

In this paper we have demonstrated the effectiveness of a SKF ensemble for systems with non-linear degradation patterns. In addition, the performance of the SKF ensmeble on NASA's C-MAPSS dataset has shown improvement over other methods in literature. Implementation on these simulated datasets simply serve as a proof-of-concept for the proposed method at this stage. This method has also wide applications to other prognostic situations where the system involved has more than one degradation mode. An example would be where the degradation pattern of the system changes due to external factors such as operating conditions or overhaul maintenance. In view of the range of possible applications, the authors have plans to implement the proposed method on a real-world dataset and validate its effectiveness.

ACKNOWLEDGMENT

This work was supported by the Economic Development Board Industrial Postgraduate Programme Fund under the project R-263-000-A14-592.

REFERENCES

AL-Mathami, Y. S., Everson, R., & Fieldsend, J. (2012). *Probabilistic controllled airpsace infringement tool*. Unpublished doctoral dissertation.

Baraldi, P., Mangili, F., & Zio, E. (2012). A kalman filter-based ensemble approach with application to turbine creep prognostics. *IEEE TRANSACTIONS ON RELIABILITY*, *61*.

Borguet, S., & Léonard, O. (2009). Coupling principal component analysis and kalman filtering algorithms for on-line aircraft engine diagnostics. *Control Engineering Practice*, *17*(4), 494–502.

Heimes, F. (2008). Recurrent neural networks for remaining useful life estimation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 231–238.

Le Son, K., Fouladirad, M., & Barros, A. (2012). Remaining useful life estimation on the non-homogenous gamma with noise deterioration based on gibbs filtering: A case study. In *Prognostics and health management (phm), 2012 ieee conference on* (pp. 1–6).

Maclin, R., & Opitz, D. (2011). Popular ensemble methods: An empirical study. *arXiv preprint arXiv:1106.0257*.

Murphy, K. P. (1998). *Switching kalman filters* (Tech. Rep.). Citeseer.

Peel, L. (2008). Data driven prognostics using a kalman filter ensemble of neural network models. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Puskorius, G. V., & Feldkamp, L. A. (1991). Decoupled extended kalman filter training of feedforward layered networks. In *Neural networks, 1991., ijcnn-91-seattle international joint conference on* (Vol. 1, pp. 771–777).

Re, M., & Valentini, G. (2011). Ensemble methods: a review.

Saxena, A., & Goebel, K. (2008). *Phm08 challenge data set, nasa ames prognostics data repository.* Moffett Field, CA. Retrieved from `[http://ti.arc.nasa.gov/project/prognostic-data-repository]`

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–9).

Singhal, S., & Wu, L. (1989). Training feed-forward networks with the extended kalman algorithm. In *Acoustics, speech, and signal processing, 1989. icassp-89., 1989 international conference on* (pp. 1187–1190).

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC Press.

A Thermodynamic Entropy Based Approach for Prognosis and Health Management

Anahita Imanian[1], Mohammad Modarres[2]

[1,2]*University of Maryland, Department of Mechanical Engineering, Center for Risk and Reliability, College Park, Maryland, 20742, USA*

aimanian@umd.edu
modarres@umd.edu

ABSTRACT

Data-driven stochastic and probabilistic methods that underlie reliability prediction and structural integrity assessment remain unchanged for decades. This paper provides a method to explain the Prognostics and Health Management (PHM) in terms of fundamental concepts of science within the irreversible thermodynamic framework. The common definition of damage, which is widely used to measure the reduction of reliability over time, is based on observable markers of damage at different geometric scales. Observable markers are typically based on evidences of any change in the physical or spatial properties or the materials, and exclude unobservable and highly localized damages. Thermodynamically, all forms of damage share a common characteristic: "energy dissipation". Energy dissipation is a fundamental measure of irreversibility that within the context of non-equilibrium thermodynamics is quantified by "entropy generation". The definition of damage in the context of thermodynamics allows for incorporation of all underlying dissipative processes including unobservable markers of damage. Using a theorem relating entropy generation to energy dissipation associated with damage producing failure mechanisms, this paper presents an approach that formally describes and measures the resulting damage.

Having developed the approach to derive the damage over time, one could assess the health of structures and components subject to known degradation processes. This paper presents a prognostic approach on the basis of thermodynamically derived cumulative damage, whereby the thermodynamic entropy, as a broad measure of damage, is assessed.

Anahita Imanian et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The definition of damage due to the physical mechanisms varies at different geometric and scales. For example, the definition of fatigue damage can vary from nano-scale through the macro-scale. At the atomic level the grain boundary is a likely location where atoms are more loosely packed. At the micro-scale damage is the accumulation of micro-stresses in the neighborhood of cracks. At the meso-scale level, damage might be defined as growth and coalescence of micro-cracks to meso-cracks. However, measuring damage is subject to the physically measurable variables (i.e., observable marker) when dealing with specific failure mechanisms. For example, in the fatigue mechanism material density, change of hardness, module of elasticity, accumulated number of cycles-to-failure, and crack length may be used as "observable markers" that measure the damage. Therefore, defining a consistent and broad definition of damage is necessary and plausible. To reach this goal, we elaborate on the concept of material damage within the thermodynamic framework.

Thermodynamically, all forms of damage share a common characteristic, which is the dissipation of energy. In thermodynamics, dissipation of energy is the basic measure of irreversibility, which is the main feature of the degradation processes in materials (Tang & Basaran, 2003). Chemical reactions, release of heat, diffusion of materials, plastic deformation, and other means of energy production involve dissipative processes. In turn, dissipation of energy can be quantified by the *entropy generation* within the context of irreversible thermodynamics. Therefore, dissipation (or equivalently entropy generation) can be considered as a substitute for characterization of damage. We consider this characterization of damage highly general, consistent and scalable.

The common practice in damage analysis and prediction of structural life and integrity is based on the traditional generic handbook-based reliability prediction methods, data driven prognostics approaches and Physics-of-Failure (PoF)

methods. The traditional generic handbook-based reliability prediction methods such as those advocated in MIL-HDBK-217F (U. S. Department of Defence, 1965), Telcordia SR-332 (Telcordia Technologies, 2001), and FIDES (FIDES Guidance Issue, 2004) rely on the analysis of field data (with incoherent operating and environmental conditions), with the assumption that the failure rates are constant. Numerous studies have shown that these methods cause misleading and inaccurate results and can lead to poor design and incorrect reliability prediction and operating decisions (IEEE Standard 1413, 1998; IEEE Standard 1413.1, 2002). The PoF models (Manson, 1996; Norris & Landzberg, 1969; Bayerer, Hermann, T. Licht, Lutz, & Feller, 2008; Shi & Mahadevan, 2001; Harlow & Wei, 1998) are more rigorous in terms of employing the specific knowledge of products, such as failure mechanism, material properties, loading profile and geometry. However, such empirical methods are limited to simple failure mechanisms and are hard to model when multiple competing and common cause failure mechanisms are involved. Finally, the data driven methods such as neural networks (Byington, Watson, & Edwards, 2004), decision tree classifiers (Schwabacher & Goebel, 2007) and Bayesian techniques (Bhangu, Bentley, Stone, & Bingham, 2005) do not capture the difference between failure modes and mechanisms, although they can obtain the complex relationship and degradation trend in the data without the need for the particular product characteristics such as degradation mechanism or material properties. Moreover, these methods require rich historical knowledge of materials and structural degradation behavior that may not always be available.

In this paper, we introduce an entropy-based prognostic approach to predict the Remaining Useful Life (RUL) of components and structures. This approach is based on the second law of thermodynamics and defines entropy as a more consistent measure of damage. As compared to other existing PoF or fusion prognostics methods (Held, Jacob, Nicoletti, Scacco, & Poech, 1999; Ciappa, 2002; Cheng & Pecht, 2009), this approach captures the effect of multiple failure mechanisms[1], more effectively. Moreover, the results of entropy approach are favorably used in fracture mechanics, fatigue damage analysis (Bryant, Khonsari, & Ling, 2008; Tang & Basaran, 2003) and tribological processes such as friction and wear (Amiri & Khonsari, 2010; Nosonovsky & Bhushan, 2009). Furthermore, it is a powerful technique to study the synergistic effects arising from interaction of multiple processes (Amiri & Khonsari, 2010).

The remainder of this paper is organized as follows. Section 2 describes our construction of the entropy model. Section 3 describes an entropic based framework for prognosis. Section 4 provides a case study which explores the application of the proposed prognostics framework, and section 5 offers concluding remarks.

2. TOTAL ENTROPY PRODUCED IN A SYSTEM

Consistent with the second law of thermodynamics, entropy does not obey a conservation law. Therefore, it is essential to relate the entropy not only to the entropy crossing the boundary between the system and its surroundings, but also to the entropy produced by the processes taking place inside the system. Processes occurring inside the system may be reversible or irreversible. Reversible processes inside a system may lead to the transfer of the entropy from one part of the system to other parts of the interior, but do not generate entropy. Irreversible processes inside a system, however, result in generation of the entropy, and hence in computing the entropy they must be taken into account.
Using the second law of thermodynamic, it is possible to express the variation of total entropy flow per unit volume, dS, in the form of

$$dS = d^r S + d^d S \tag{1}$$

where, S is defined for a domain g by means of specific entropy, s, per unit mass as $S = \int_g \rho s dV$, and the super scribes r and d represent the reversible and irreversible part of the entropy, respectively. The term $d^r S$ is the entropy supplied to the system by its surroundings through transfer of mass and heat (e.g., in an open system where wear and corrosion mechanisms occur). The rate of exchanged entropy is obtained as

$$\frac{d^r S}{dt} = -\int^{\Omega} J_s . n_s dA \tag{2}$$

where, J_s is a vector of the total entropy flow per unit area, crossing the boundary between the system and its surroundings, and n_s is a normal vector. Similarly, $d^d S$ is the entropy produced inside of the system, which can be obtained from Eq. 3

$$\frac{d^d S}{dt} = \int^{V} \sigma dV \tag{3}$$

where, σ is the entropy generation per unit volume per unit time. The second law of thermodynamics states that $d^d S$ must be zero for reversible transformations and positive ($d^d S > 0$) for irreversible transformations of the system.
The balance equation for entropy shown in Eq. 4 can be derived using the conservation of energy and balance equation for the mass.

$$\frac{ds}{dt} + \nabla J_s = \sigma \tag{4}$$

[1] Particularly, in contrast with the empirically-based PoF approach which considers only the most predominant failure mechanisms, the definition of damage in the context of the entropic approach allows for the incorporation of all underlying dissipative processes. For example, in the case of corrosion-fatigue, both stress and electrochemical affinity of the oxidation-reduction electrode reaction ($Me \Leftrightarrow Me^{z+} + ze$) of a metal are considered.

This gives us an explicit expression for total entropy in terms of reversible and irreversible processes as (De Groot & Mazor, 1962; Kondepudi & Prigogine, 1998)

$$\frac{ds}{dt} = -\nabla.\left(\frac{J_q - \sum_{k=1}^{n}(c_m\psi + \mu_k)J_k}{T}\right) + \frac{1}{T^2}J_q.\nabla T - \Sigma_{k=1}^{n}J_k\left(\nabla\frac{\mu_k}{T}\right) + \frac{1}{T}\tau:\dot{\epsilon_p} + \frac{1}{T}\Sigma_{j=1}^{r}v_j\,A_j + \frac{1}{T}\Sigma_{m=1}^{h}c_mJ_m(-\nabla\psi) \quad (5)$$

where, T is the temperature, μ_k the chemical potential, J_q the heat flux, J_k the diffusion flow, J_m any fluxes resulting from external fields (magnetic and electrical) such as electrical current, v_i the chemical reaction rate, τ the stress tensor, $\dot{\epsilon_p}$ the plastic strain rate tensor, $A_j = -\Sigma_{i=1}^{u}\mu_i\nu_{ji}$ the chemical affinity or chemical reaction potential difference, ψ the potential of the external field such as electrical potential difference, and c_m the coupling constant. External forces may be resulted from different factors including electrical field, magnetic field, gravity field, etc., where the corresponding fluxes are electrical current, magnetic current and velocity. For example, in the case of an electric field, $E = -\nabla\psi$ is the electric potential, $I = \Sigma_{m=1}^{h}c_mJ_m$, the current density and $c_m = Fz_m$, where F is the Faraday constant and z_m is the number of ions. Each term in Eq. 5 is derived from the various mechanisms involved, which define the macroscopic state of the complete system.

By comparing Eq. 5 with Eq. 4 we can make the identifications as

$$J_s = \frac{J_q - \sum_{k=1}^{n}(c_m\psi + \mu_k)J_k}{T} \quad (6)$$

$$\sigma = \frac{1}{T^2}J_q.\nabla T - \Sigma_{k=1}^{n}J_k\left(\nabla\frac{\mu_k}{T}\right) + \frac{1}{T}\tau:\dot{\epsilon_p} + \frac{1}{T}\Sigma_{j=1}^{r}v_j\,A_j + \frac{1}{T}\Sigma_{m=1}^{h}c_mJ_m(-\nabla\psi) \quad (7)$$

where, Eq. 6 shows the entropy flux resulted from heat and material exchange. Eq. 7 represents the total energy dissipation terms from the system that from left to the right include heat conduction energy, diffusion energy, mechanical energy, chemical energy, and external force energy. Eq. 7 is fundamental to non-equilibrium thermodynamics, and represents the entropy generation σ as the bilinear form of forces and fluxes as

$$\sigma = \Sigma_{i,j}X_iJ_i(X_j);\ (i, j=1, \ldots, n) \quad (8)$$

It is through this form that the contribution from the applicable thermodynamic forces and fluxes are expressed. When multiple failure mechanisms are involved in a degradation process such as corrosion fatigue, summing the contributions of the mechanical and electrochemical processes, one can write the total entropy generation for combined effect of plastic deformation and anodic and catholic dissolution as:

$$T\sigma = \tau:\dot{\epsilon_p} + \tilde{A}i_{corr} \quad (9)$$

where $\tilde{A}$ is the electrochemical potential losses (over-potential) (Imanian & Modarres, 2014). Additionally, using forces and fluxes enables one to take into account complex loading scenarios and operating conditions in computing entropy produced in degradation processes.

3. RUL PREDICTION USING ENTROPY AS AN INDEX OF DAMAGE

It was stated earlier that damage caused through a degradation process could be viewed as the consequence of dissipation of energies that can be measured and expressed by entropy such that:

Damage ≡ Entropy

In the earlier discussion in this paper it was shown (Eq. 5) that one could express the total entropy per unit time per unit volume for individual dissipation processes resulting from the corresponding failure mechanisms. Therefore, the evolution trend of the damage, D, is obtained from

$$D|t \sim \int_0^t [\sigma|X_i(u), J_i(u)]du \quad (10)$$

where, $D|t$ is the monotonically increasing cumulative damage starting at time *t* from a theoretically zero value or practically some initial damage value. In this study, the evaluation of damage is performed relative to the initial damage value. The initial damage can be calculated using the correlation between the rate of damage and damage at different stage of degradation (Liakat & Khonsari, 2014).

When D reaches a predefined (often subjective) level of endurance, it may be assumed that beyond that point the component or structure will fail. It is worth to note that failure in this context is the point when an item becomes effectively nonfunctional (but possibly still operational) – i.e., failure happens when the item is no longer meeting its functionality requirements (e.g., acceptable performance level or endurance limit such as a given level of thermodynamic efficiency). The rate of entropy or damage can vary according to the type of degradation. However, damage in the system mounts up over time. For example, in the case of fatigue crack closure, while the crack as an observable marker of damage disappears, causing damage rate decrease, the damage accumulation keeps rising as unobservable markers of damage such as loading asymmetry, hardening properties, residual stresses and loading ratio increase (Romaniv, Nikiforchin, & Andrusiv, 1983).

Material, environmental, operational and other types of variability in degradation forces impose uncertainties on the

cumulative damage, D. Existence of any uncertainties about the parameters and independent variables in this thermodynamic-based damage model leads to a time-to-failure distribution. Imanian et al. showed how such a distribution and corresponding reliability function can be derived from the thermodynamic laws rather than estimated from the observed time to failure histories (Imanian & Modarres, 2014).

Currently, most of the health management of components and structures is based on reliability analysis and maintenance scheduling. However, in many cases this is neither sufficient nor efficient because each of these components can undergo different life cycles and hence different aging. Therefore, if maintenance or replacement is done solely based on reliability analysis, in most circumstances the components will either be abandoned before they have reached their end of life, or worse, they will fail before their scheduled replacement.

Prognostics and health management modeling approaches are used to reduce the costs of the physics based propagation damage. The techniques included in the PHM provide warnings before failures happen; they also optimize the maintenance schedule, reduce life cycle cost of inspection, and improve qualification tests assisted in design and manufacturing. Prognostics and health management modeling methods are implemented through three stages of diagnostics, prognostics, and health management. Diagnostics techniques identify the operational states of a working component or a structure. These techniques use statistics features such as mean, standard deviation, Mahalanobis distance and Euclidean distance of a component's degradation operating data (e.g. temperature, current, voltage, acoustic signals) to find out if the component is in a healthy condition or not regarding the feature's level degradation (Schwabacher & Goebel, 2007; Bock, Brotherton, Grabill, Gass, & Keller, 2006; Fraser, Hengartner, Vixie, & Wohlberg, 2003).

Prognostics methods provide information about the performance and RUL of components by modeling degradation propagation. These methods rely on the condition of the data which can roughly be divided into data driven based models and PoF based models. PoF based prognostics methods employ knowledge of products life cycle loading profile, failure mechanisms, geometry, and material properties. However, using PoF models is challenging because these methods are based on the interactions among multiple failure mechanisms which are not easy to analyze. Data driven based models are able to obtain the complex relationship and degradation trend in the data without the need for the particular product characteristics such as degradation mechanism or material properties (Amin, Byington, & Watson, 2005; Byington, Watson, & Edwards, 2004; Roemer, Ge, Liberson, Tandon, & Kim, 2005; Goebel, Saha, & Saxena, 2008). However, they cannot capture the difference between failure modes and mechanisms.

Since entropy function includes all of the failure mechanisms' dissipative energies when multiple competing and common cause failure mechanisms are involved, using it as a damage parameter for diagnosis and prognostics is more favorable in comparison with the PoF models and data driven models which merely rely on the most predominant failure mechanisms and the statistical analysis, respectively. What follows presents an entropy based prognostics method for RUL prediction. The proposed prognostics framework is depicted in Figure 1.

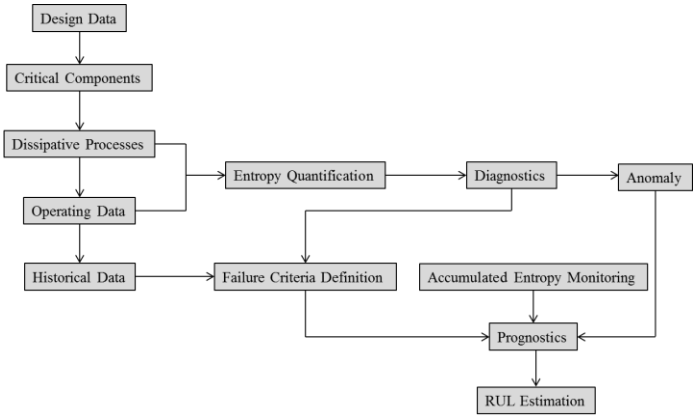


Figure 1. RUL prediction by entropy based prognostic method.

According to this framework the entropic base prognostics method can be implemented in four steps. First, the dissipative processes and associated data in the critical components under aging are determined. The identification of these processes and relevant parameters can be aided by failure modes, mechanisms, and effects analysis (FMMEA) which identifies the potential failure mechanisms for products, under certain environmental and operating conditions. The entropy as a parameter of damage which includes all the interactive failure mechanisms is quantified then.

The second step is to extract the features of the monitored entropy data and compare them with the healthy baseline data features to detect anomalies. The traditional diagnostic approaches are mainly designed for stationary and known operating conditions. The problem of a fault diagnosis under fluctuating load and operating conditions has been successfully addressed by methods such as order tracking method (Stander & Heyns, 2005), instantaneous power spectrum statistical analysis (Bartelmus & Zimroz, 2009), and diagnosis algorithms such as clustering algorithms (Schwabacher & Goebel, 2007; Vapnik, 1995; He & Wang, 2007).

Because entropy as a parameter of degradation includes all observable damage markers (cracks, wear debris and pit

densities) and unobservable damages such as subsurface dislocations, slip and micro-cavities, definition of a single failure threshold might not be possible due to long stretch of damage measurement from nano-scale to macroscopic scale. In this case, the cumulative damage and alternatively entropy endurance level can be estimated through the measurement of certain observable damage markers. The correlation between the observable damage markers and entropy, justified by several studies (Naderi, Amiri, & Khonsari, 2010; Bryant, Khonsari, & Ling, 2008), enables the definition of failure threshold on the basis of observable markers. In the other word, the damages grow, coalesce and eventually the weakest link among all coalesces damages manifests itself as an observable damage which causes failure.

Additionally, records of the entropy data from historical data can be used to obtain the entropy to failure values. Entropy, as a thermodynamic state function is independent of the path to failure (loading values, frequency and geometry) and provides an overall constant failure criterion (Kondepudi & Prigogine, 1998; Bryant, Khonsari, & Ling, 2008).

The third step is to use an appropriate prognostics approach using entropy as an index of damage. Some of the conventional methods used for prognostics are artificial neural network (Byington, Watson, & Edwards, 2004; Amin, Byington, & Watson, 2005), fuzzy logic (Amiri & Khonsari, 2010), wavelet theory (Roemer, Ge, Liberson, Tandon, & Kim, 2005), support vector machine (Vapnik, 1995), relevance vector machine (Tipping, 2000), Bayesian methods (like Kalman filter and Particle filter (Arulampalam, Maskell, Gordon, & Clapp, 2002)), time series analysis (Kumar & Pecht, 2007) and PoF based prognostics models. The application of these methods depends to the complexity of accumulated entropy signal from two extremes of periodic and purely random signal.

The fourth and final step is RUL prediction. Remaining useful life is defined as the time when the entropy meets the failure criteria. There are different techniques for RUL estimation using data driven methods. For example one approach uses a pattern matching technique on data to estimate the RUL. Another strategy estimates the RUL indirectly by estimating damage trend, performing an appropriate extrapolation to the damage trend, and the calculation of RUL from the intersection of the extrapolated damage and the failure criteria (Schwabacher & Goebel, 2007). In comparison with the end of life prediction from entropy trend, the conventional RUL prediction methods are based on a damage mechanism with different failure mechanisms. These various failure mechanisms with different failure criteria and parameters' trends have various RULs which needs them to be prioritized accordingly (Cheng & Pecht, 2009).

Generally speaking, using entropy as a damage parameter has various advantages. The entropy based prognostics method is capable of shortening the prognostics procedure by isolating the damage parameter to entropy which includes multiple degradation mechanisms. It offers a science based foundation for prognostic methods which could combine with the conventional data driven techniques, as compared to the methods suggested by previous studies such as fusion prognostic approach suggested by Cheng et al (Cheng & Pecht, 2009). Furthermore, it uses a constant failure threshold and suggests a straightforward process to predict RUL (Amiri & Khonsari, 2010).

4. CASE STUDY

The entropy based prognostics approach was employed to obtain the remaining useful life of the AL7075-T651 coupons subjected to fatigue loading, using an MTS servo-hydraulic uni-axial load frame, from Ontiveros et al. experimental results (Ontiveros, 2013). Geometries of the coupons used are shown in Figure 2. All tests were performed at peak stress of 248 MPa with load ratio of 0.1 and frequency of 2Hz. Since the focus of Ontiveros et al. study was crack initiation, so most of experiments were stopped when, a crack was detected at the notch by visual inspection.

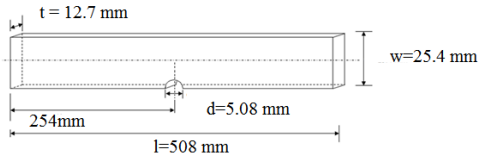


Figure 2. Al7075-651 edge notch specimen.

The formulation for entropy generation using Eq. 7 can be derived as

$$\sigma = \frac{\tau : \dot{\varepsilon_p}}{T} + \frac{1}{T} Z\dot{D} + \frac{1}{T^2} J_q \cdot \nabla T \qquad (11)$$

where, Z is the elastic energy release rate and $\dot{D}$ is the damage rate variable.

In Eq. 11, the first two terms can be captured directly from the hysteresis loop as depicted in Figure 3. In Figure 3, the largest area represents the energy dissipated due to plastic deformation. The remaining portion represents the energy dissipation as a result of elastic damage which can be observed as degradation of the Young's modulus (Lemaitre & Chaboche, 1990).

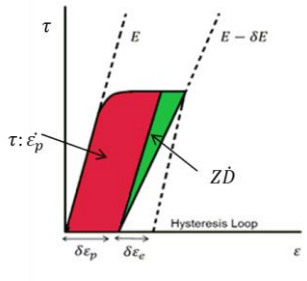


Figure 3. Hysteresis Energy (Reproduced from (Ontiveros, 2013)).

Results of Ontiveros et al. analysis showed that when compared to the plastic and elastic energy dissipations the fraction of the entropy generation due to heat conduction is considered to be negligible. Therefore, the third term does not take into account in the entropy calculation.

The prognostic framework implemented in this study involves the measurement of parameters included in the Eq. 11 and using the entropy as a parameter to be monitored. Specific Mahalanobis Distance (MD) is used as a diagnostic threshold which triggers the prediction. Once an anomaly is detected, the Particle Filter (PF) procedure is initiated for time to failure prognostic. The failure threshold in this approach is the mean of the failure threshold of the 3 samples considered as training samples.

4.1. Anomaly Detection

To obtain the anomaly threshold for every entropy data point, the MD values are calculated based on the distance between healthy and anomalous data. Then, the calculated MD values are transformed into a normal distribution using the Box-Cox transformation method (Box & Cox, 1964). After that, a detection threshold is quantified upon the mean and standard deviation of the transformed healthy MD data. The calculations are repeated for every test data, and anomaly is marked for every test point which goes beyond the detection threshold.

To implement the MD, entropy data are divided into two categories: (i) healthy data and (ii) test data. The observations between 4000 and 5500 cycles were classified as healthy data and the whole set of observations was considered as test data. The number of observations recorded for entropy parameter is denoted by k, where $k = 1, 2, \dots n$. S_k is the values of entropy at cycle k. Each individual observation of entropy data vector was normalized using the mean, $\overline{S_h}$, and standard deviation, D_h, from the healthy entropy data using Eq. 12.

$$Y_k = \frac{\mathrm{S} - \overline{S_{h_k}}}{D_{h_k}} \tag{12}$$

The MD values were computed by using Eq. 13.

$$MD_k = Y_k^T C^{-1} Y_k \tag{13}$$

Where C is the correlation matrix which can be obtained by

$$C = \frac{1}{n-1}\sum_{k=1}^{n} Y_k Y_k^T \tag{14}$$

Since the healthy MD values were found to not follow a normal distribution, the Box-Cox power transformation was employed to convert the healthy MD values into a normal distribution. This transformation allows for the use of statistical mean to determine the healthy or unhealthy conditions of the data. The Box-Cox transformation is defined by Eq. 15, where $MD(\lambda)$ is the transformed vector, MD is the original vector, and λ the transformation parameter.

$$\begin{aligned} MD(\lambda) &= \frac{MD^{\lambda} - 1}{\lambda} && \lambda \neq 0 \\ MD(\lambda) &= \ln(MD) && \lambda = 0 \end{aligned} \tag{15}$$

The mean and standard deviation of the transformed healthy values were used to define the threshold for anomaly detection as $\overline{S_{h_k}} + 3D_h$. When a transformed test $MD(\lambda)$ values (based on the Box-Cox transformation using parameter λ learned from the healthy data) crosses this threshold, an anomaly was considered to have occurred.

4.2. Particle Filter Prediction

By choosing the entropy data as a feature of damage, Bayesian method can be used to update the parameters of the model and the age predictions. Bayesian approaches provide a general rigorous method for dynamic state estimation problems. The idea is to build a Probability Density Function (PDF) of the system states based on all available information. Particle Filter (PF) is a method for implementing a recursive Bayesian filter using Monte Carlo simulations. Particle Filter (PF) approximates the model parameters' PDF by a set of particles sampled from the distribution and a set of associated weights denoting probability masses (Arulampalam, Maskell, Gordon, & Clapp, 2002).

In particle filter method, the particles are generated and recursively updated by process model shown in Eq. 16, a measurement model depicted in Eq. 17 and an *a priori* estimate of the state PDF.

$$\vec{x}_k^i = f_k\left(\vec{x}_{k-1}^i, \Omega_{\mathrm{k-1}}\right) \tag{16}$$

$$\vec{y}_k = H_k(\vec{x}_k, \Psi_k) \tag{17}$$

where, Ω_k and Ψ_k are the system and measurement noises, respectively. Defining the model parameter vector at cycle k as $\vec{x}_k = [a_1,\ a_2, \dots,\ a_n]$ and damage level measurements as $\vec{y}_k = [S_0,\ S_1, \dots.,\ S_m]$, the particle filter is implemented by initiating the state of the system by a set of particles $\vec{x}_0^i$, where $i = 1, 2, \dots, N_s$.

If $\{(w_k^i, \vec{x}_k^i)\}$ denotes a random measure that characterizes the posterior PDF, $p(x_{0:k}|y_{1:k})$ (where $\{x_{0:k}^i, i = 0, , \dots N_s\}$, is a set of support points with associated weights $\{w_k^i, i = 0, , \dots N_s\}$, normalized such that $\sum_{i=1}^{N_s} w_k^i = 1$) the posterior density at cycle k can be approximated as

$$p(\vec{x}_{0:k}|\vec{y}_{1:k}) = \Sigma_{i=1}^{N_s} w_k^i \delta(\vec{x}_{0:k} - \vec{x}_{0:k}^i) \quad (18)$$

where, $\vec{x}_{0:k}$ and $\vec{y}_{1:k}$ are the set of all states and measurements up to cycle k. *Sampling importance resampling* is a commonly used algorithm to attribute importance weight, w_k^i, to each particle, i,

$$w_k^i = \frac{p(\vec{y}_{1:k}|\vec{x}_k^i)p(\vec{x}_k^i)}{\pi(\vec{x}_k^i|\vec{y}_{1:k})} \quad (19)$$

The posterior PDF is then calculated by

$$w_k^i = w_{k-1}^i \frac{p(\vec{y}_k|\vec{x}_k^i)p(\vec{x}_k^i|\vec{x}_{k-1}^i)}{\pi(\vec{x}_k^i|\vec{x}_k^i, \vec{y}_{1:k})} \quad (20)$$

where the importance distribution $\pi(\vec{x}_k^i|\vec{x}_k^i, \vec{y}_{1:k})$ is approximated by $p(\vec{x}_k^i|\vec{x}_{k-1}^i)$ (Arulampalam, Maskell, Gordon, & Clapp, 2002).

4.3. Remaining Useful Life Prediction

To tie in the aforementioned technique, namely PF approach, with the entropic based prognosis, the system model can be represented by a regression model, based on accumulated entropy values, S', from experimental data analysis

$$S'_k = a_{1_k} k + a_{0_k} \quad (21)$$

which delivers a good fit for the entropy increment of Al specimens subjected to fatigue mechanism. Here, k is the cycle number, and a_1 and a_0 are the model parameters subjected to a Gaussian error as

$$\begin{aligned} a_{0_k} &= a_{0_{k-1}} + \omega_{a_0} \\ \text{where: } \omega_{a_0} &\sim N(0, std_{a_0}) \\ a_{1_k} &= a_{1_{k-1}} + \omega_{a_1} \\ \text{where: } \omega_{a_1} &\sim N(0, std_{a_1}) \end{aligned} \quad (22)$$

Given a series of measured entropy values, S', subjected to a Gaussian noise, $N(0, std)$ with zero mean and standard deviation std, as

$$\begin{aligned} S'_k &= a_{1_k} k + a_{0_k} + \psi \\ \text{where: } \psi &\sim N(0, std) \end{aligned} \quad (23)$$

the PF technique enables the estimation of the model parameters (a_1 and a_0) where in the updating process, N_s samples are used to approximate the posterior PDF. Each sample denotes a candidate for the model parameter vector $\vec{x}_k^i = [a_{0_k}, a_{1_k}]$, $i = 1, 2, \dots, N_s$, so the prediction of S' would have N_s possible trajectories with the corresponding importance weight w_k^i. The h^{th} steps ahead prediction of each trajectory at cycle k is calculated by

$$S'^i_{k+h} = a_{1_k}^i (k + h) + a_{0_k}^i \quad (24)$$

The estimated PDF of the entropy prediction can be obtained by

$$p(S'_{k+h}|S'_{0:k}) = \sum_{i=1}^{N_s} \omega_k^i \delta(S'_{k+h} - S'^i_{k+h}) \quad (25)$$

Since the failure threshold is defined as the mean of entropy to failure of training entropy data taken from 3 samples, S'_f, the remaining useful life probability estimation, R_k^i, of the i^{th} trajectory at cycle k can be obtained by solving the following equation

$$S'_f = a_{1_k}^i (k + R_k^i) + a_{0_k}^i \quad (26)$$

The PDF of the RULs at cycle k can be approximated by

$$p(R_k|S'_{0:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(R_k - R_k^i) \quad (27)$$

4.4. Prognostics Results

Using the MD approach, anomalies were identified when the transformed MD threshold of the test entropy data crosses the anomaly detection threshold. Once the anomaly was detected, the PF algorithm was initiated to predict RUL. The system model used for particle filter prediction follows Eq. 23. The initial values of the model parameters were obtained from the least square regression for each specimen, using the healthy interval of the data. Figure 4 shows prediction results for specimen number 6. The yellow zone shows the shape of RUL probability density function estimation after anomaly criteria detected.

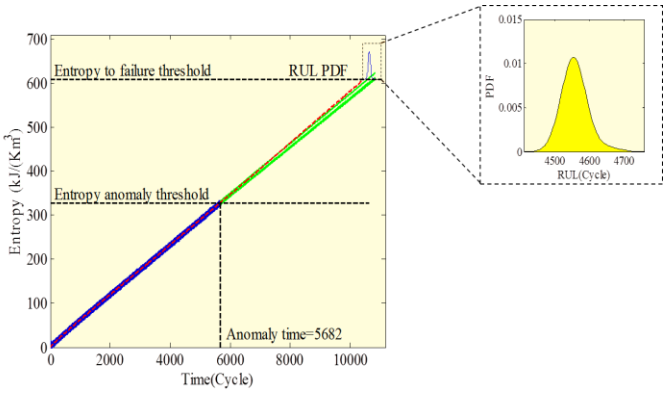


Figure 4. Predicted failure distribution at the time of anomaly detection for specimen number 6.

The same procedure applied to the 6 remaining specimens. The values for the mean of the predicted RULs and actual RULs are shown in Table 1. The error between mean of estimated RULs and actual RULs falls in the reasonable range of 4% to 18%.

Table 1. Comparisons of the actual and estimated RULs

Sample no.	RUL_{actual}(Cyc)	Mean($RUL_{estimated}$(Cyc))	Error
1	2829	2635.5	7%
2	3827	3563	7%
3	11165	10696.5	4%
4	1987	1621	18%
5	1018	835.5	17%
6	4792	4596	4%
7	3604	3444	4%

5. CONCLUSION

This paper presents an effort to use a thermodynamic framework, using entropy generation as a measure of damage, to assess RUL of a component or structure. It introduces a unified measure of damage in terms of energy dissipations for multiple irreversible processes with reference to physically measurable quantities. As compared to other existing PoF, data driven, or fusion prognostics methods, entropic-damage models capture the effect of multiple competing and common-cause failure mechanisms. The RUL predicted by this method includes the effect of all failure mechanisms and unlike conventional RUL prediction methods, where various RULs correspond to different failure mechanisms, it provides a unified RUL.

This paper also demonstrates a case study for implementation of an entropy-based prognostics method. Particle filter is applied to update the states of the model, reduce uncertainties and predict the RUL probability distribution function. The proposed method provides satisfactory RUL predictions.

While the entropy method proves to be theoretically more relevant for reliability analysis, its advantages remain to be explored practically. One practice in this regard is the authors' current project on introducing the entropy growth rate as a degradation parameter to the corrosion-fatigue mechanisms in materials.

ACKNOWLEDGEMENT

This work is part of an ongoing research through grant number N000141410005 from the Office of Naval Research (ONR).

References

Amin, S., Byington, C., & Watson, M. (2005). Fuzzy Inference and Fusion for Health State Diagnosis of Hydraulic Pumps and Motors. *Proceeding of the Annual Meeting of the North American Fuzzy Information Processing Society*, (pp. 13-18). Detroit, MI.

Amiri, M., & Khonsari, M. (2010). On the Thermodynamics of Friction and Wear - A Review. *Journal of Entropy, 12*, 1021-1049.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transaction on Signal Processing, 50*(2), 174-189.

Bartelmus, W., & Zimroz, R. (2009). A New Feature for Monitoring the Condition of Gearboxes in Non–stationary Operating Conditions. *Journal of Mechanical Systems and Signal Processing, 23*(12), 1528-1534.

Bayerer, R., Hermann, T. Licht, T., Lutz, J., & Feller, M. (2008). Model for Power Cycling Lifetime of IGBT Modules - Various Factors Influencing Lifetime. *Integerated Power Electronics Systems Conference*, (pp. 11-13). Nuremberg, Germany.

Bhangu, B. S., Bentley, P., Stone, D. A., & Bingham, C. M. (2005). Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles. *IEEE Transactions on Vehicular Technology, 54*(3), 783-798.

Bock, J. R., Brotherton, T., Grabill, P., Gass, D., & Keller, J. R. (2006). On False ALarm Mitigation. *IEEE Aerospace Conference*, (pp. 1-16). New York, USA.

Box, G., & Cox, D. (1964). An Analysis of Transformations. *Journal of Royal Statistical Society, 26*, 211-252.

Bryant, M. D., Khonsari, M. M., & Ling, F. F. (2008). On the Thermodynamics of Degradation. *Journal of Royal Society, 464*, 2001-2014.

Byington, C., Watson, M., & Edwards, D. (2004). Data-Driven Neural Network Methodology to Remaining Life Predictions for Aircraft Actuator Components. *Proceedings of the IEEE Aerospace Conference*, (pp. 3581-3589). New York, USA.

Cheng, C., & Pecht, M. (2009). A Fusion Prognostics Method for Remaining Useful Life Prediction of Electronic Products. *5th annual IEEE Conference on Automation Science and Engineering*, (pp. 22-25). Bangalore, India.

Ciappa, M. (2002). Selected Failure Mechanisms of Modern Power Modules. *Journal of Microelectronic Reliability, 42*, 653-667.

De Groot, S. R., & Mazor, P. (1962). *Non-Equilibrium Thermodynamics.* New York: John Wiley & Sons.

FIDES Guidance Issue. (2004). *Reliability Methodology for Electronic Systems.* FIDES Group.

Fraser, A., Hengartner, N., Vixie, K., & Wohlberg, B. (2003). Incorporating Invariants in Mahalanobis Distance Based Classifiers: Application to Face Recognition. *International Joint Conference on Neural Networks*, (pp. 3118-3123). Portland, USA.

Goebel, K., Saha, B., & Saxena, A. (2008). A Comparison of Three Data-Driven Techniques for Prognostics. *62nd Meeting of the Society for Machinery Failure Prevention Technology.* Virginia Beach, VA, USA.

Harlow, D. G., & Wei, R. P. (1998). A Probability Model for the Growth of Corrosion Pits in Aluminum Alloys Induced by Constituent Particles. *Journal of Engineering Fracture Mechanics, 59*, 305-325.

He, Q. P., & Wang, J. (2007). Fault Detection Using K-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. *IEEE Transaction of Semiconductor Manufacturers, 20*, 345-354.

Held, M., Jacob, P., Nicoletti, G., Scacco, P., & Poech, M. H. (1999). Fast Power Cycling Test of IGBT Modules in Traction Application. *International Journal of Electronics, 86*(10), 1193-1204.

IEEE Standard 1413. (1998). *IEEE Standard Methodology for Reliability Prediction and Assessment for Electronic Systems and Equipment.*

IEEE Standard 1413.1. (2002). *IEEE Guide for Selecting and Using Reliability Predictions Based on IEEE 1413.*

Imanian, A., & Modarres, M. (2014). A Science-Based Theory of Reliability Founded on Thermodynamic Entropy. *Probablistic Safty Assessment and Managment Conference.* Honolulu, Hawaii, USA.

Kondepudi, D., & Prigogine, I. (1998). *Modern Thermodynamics: From Heat Engines to Dissipative Structures.* England: Jhon Wiley & Sons.

Kumar, S., & Pecht, M. (2007). Health Monitoring of Electronic Products Using Symbolic Time Series Analysis. *Artificial Intelligence for Prognostics, AAAI Fall Symposium Series.* Arlington, VA.

Lemaitre, J., & Chaboche, J. L. (1990). *Mechanics of solid materials.* Cambridge: Cambridge University Press.

Liakat, M., & Khonsari, M. (2014). An Experimental Approach to Estimate Damage and Remaining Life of Metals under Uniaxial Fatigue Loading. *Journal of Material & Design, 57*, 289-297.

Manson, S. S. (1996). *Thermal Stress and Low Cycle Fatigue.* New York: McGraw-Hill.

Naderi, M., Amiri, M., & Khonsari, M. (2010). On the Thermodynamic Entropy of Fatigue Fracture. *Proceedings of the Royal Society A – Mathematical Physical and Engineering Sciences, 466*, 423-438.

Norris, K., & Landzberg, A. (1969). Reliability of Controlled Collapse Interconnections. *IBM Journal of Research and Development, 13*, 266-277.

Nosonovsky, M., & Bhushan, B. (2009). Thermodynamics of Surface Degradation, Self-organization, and Self-healing for Biomimetic Surfaces. *Transaction of Royal Society, 367*, 1607-1627.

Ontiveros, V. L. (2013). Strain Energy and Thermodynamic Entopy as Prognastic Measures of Crack Initiation in Aluminum. *Doctoral Dissertation*. University of Maryland, MD, USA.

Roemer, M., Ge, J., Liberson, A., Tandon, G., & Kim, R. (2005). Autonomous Impact Damage Detection and Isolation Prediction for Aerospace Structures. *Proceeding of the IEEE Aerospace Conference*, (pp. 3592-3600). New York, USA.

Romaniv, O. R., Nikiforchin, G. N., & Andrusiv, N. N. (1983). The Effect of Cracks Closure and the Estimation of Cyclic Crack Resistance of Structural Alloys. *Journal of Physico- Chemistry, Mechanics & Materials, 3*, 47-61.

Schwabacher, M., & Goebel, K. (2007). A Survey of Artificial Intelligence for Prognostics. *AAAI Fall Symposium: AI for Prognastics.*

Shi, R., & Mahadevan, S. (2001). Damage Tolerance Approach for Probabilistic Pitting Corrosion Fatigue Life Prediction. *Journal of Engineering Fracture Mechanics, 68*, 1493-1507.

Stander, C., & Heyns, P. (2005). Instantaneous Angular Speed Monitoring of Gearboxes under Non–cyclic Stationary Load Conditions. *Journal of Mechanical Systems and Signal Processing, 19*(4), 817-835.

Tang, H., & Basaran, C. (2003). A Damage Mechanics Based Fatigue Life Prediction Model for Solder Joints. *Transaction of ASME Journal of Electronic Package, 125*, 120-125.

Telcordia Technologies. (2001). *Reliability Prediction Procedure for Electronic Equipment.* NJ: Telcordia Customer Service.

Tipping, M. E. (2000). The Relevance Vector Machine. *Journal of Advances in Neural Information Processing Systems, 12*, 652-658.

U. S. Department of Defence. (1965). *Military Handbook for Reliability Prediction of Electronic Equipment, Version A.*

Vapnik, V. N. (1995). *The Nature of Statistical Learning.* Berlin: Springer.

BIOGRAPHIES

Anahita Imanian is currently a third year PhD student in the Center for Risk and Reliability at the department of Mechanical Engineering at the University of Maryland, under the supervision of Professor Mohammad Modarres. Her doctoral work explores an entropic theory of damage with applications to corrosion-fatigue structural integrity assessment. She holds an ABD PhD, and an MS in flight dynamic and control in aerospace engineering, and a BS in aerospace engineering all from Sharif University of Technology, Tehran, Iran. Her research is supported by the Office of Naval Research (ONR).

Mohammad Modarres is currently director of nuclear engineering program and center for risk and reliability at the

department of mechanical engineering at the University of Maryland. He holds a PhD from Massachusetts Institute of Technology. He has served as a consultant to several governmental agencies, private organizations and national laboratories in areas related to probabilistic risk assessment, especially applications to complex systems and processes such as the nuclear power plants. He has over 200 papers in archival journals and proceedings of conferences and three books in various areas of risk and reliability engineering.

A Low Frequency Uni-variate Model for the Effective Diagnosis and Prognosis of Bearing Signals Based Upon High Frequency Data

Jamie L. Godwin[1], Peter Matthews[1]

[1] *School of engineering and computing sciences, Science Laboratories, Durham University, South Road, Durham, DH1 3LE*

j.l.godwin@durham.ac.uk
p.c.matthews@durham.ac.uk

ABSTRACT

Prognosis of rotating machinery is of vital importance to ensure ever increasing demands of availability, reduced maintenance expenditure and increased useful life are met. However, the prognosis of bearings typically employs techniques in the frequency or time-frequency domain due to the high frequency nature of the data involved (typically >20 KHz). This data quickly becomes unmanageable in practice and often has inferior prognostic horizons in comparison to those techniques which are based upon low frequency data analysis.

This paper presents a novel methodology based upon the computation of the deviation from the empirically derived cumulative density function (CDF) of bearing data. For this purpose, the non-parametric, two sample, uni-variate Kolmogorov-Smirnov test is employed for the analysis. In particular, this paper focuses on mitigating the requirement of a-priori knowledge for bearing prognosis.

Initially, assumptions regarding the underlying structure of high frequency bearing data are explored on publically available data, and found to deviate from what would be expected.

Exploiting this, we use the non-parametric two-sample uni-variate Kolmogorov-Smirnov test to define normal operational behaviour, whilst mitigating the requirement for a-priori knowledge. This reduces the computational complexity of the system whilst having the prospect to reduce the inherent noise within the high frequency bearing signal.

Strong trends of degradation which can be used to derive prognostic maintenance conditions are observed, with sound statistical analysis performed. In particular, statistically significant degradation is found to occur 75 hours before failure occurred (representing identification at 54.2% of bearing life). Both the Kolmogorov-Smirnov D statistic and p -value are employed as health metrics to which degradation can be inferred from. A series of 4 experiments is presented, showing the versatility of the described technique and cases where the technique cannot be employed.

The technique is validated on a failed bearing and then verified on an independent, healthy bearing, and is shown to correctly identify the bearing of question in each case, enabling the prioritisation of maintenance actions which can be used to assist in reducing overall maintenance expenditure.

Jamie. L. Godwin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

With the continually reducing cost of data storage and acquisition, prognosis of critical assets is cheaper than ever. However, the effective exploitation of all this data is not trivial. With more data comes more noise, more conflicting signals, the need for new analytical techniques and the ability to process this data in real time.

As an example, storing data sampled at 20 KHz (20,480 samples per second) requires 13.5GB of data per day, equating to almost 2 billion data points. This makes the identification of degradation within the data difficult, both in automated analysis and also for human operators who can be overloaded by the quantity of data.

Although large quantities of data are collected for analysis, only a subset of this data refers to degraded or failed conditions; in some instances, even for common fault modes, less than 0.1% of the collected data can be used in analysis (Verma & Kusiak, 2011). As such, the use of cutting edge data-mining techniques for these issues is limited. However, this can be exploited through the use of statistical techniques to exploit the known normal behaviour of the data which has been collected.

Data has been identified as a key enabler of next generation maintenance methodologies - such as E-Maintenance

(Levrat et al., 2008) - due to the benefit of 5 key points (Hameed et al., 2009):

1. The ability to avoid premature breakdowns
2. Reducing the cost of maintenance
3. Enabling remote diagnosis
4. Increasing production through effective maintenance scheduling
5. Design refinement due to better quality analysis

In this work, a robust uni-variate model for the effective diagnosis and prognosis of bearings is presented. Publically available data collected by the IMS centre and made available by NASA (Lee et al., 2007) is employed to derive a sound statistical time based feature which can be used to determine asset condition. By exploiting normal operational behaviour characterised by the distribution of high frequency data, deviation from expected behaviour can be identified by empirical analysis of the cumulative density function (CDF) of the data. For this purpose, the non-parametric uni-variate Kolmogorov-Smirnov test is used to quantify the deviation from the known behaviour state to the degraded state, whilst quantifying statistically the likelihood of degradation being present.

This overcomes the current limitations of statistical pattern recognition techniques employed in prognostics and health management by empirically defining the CDF and measuring deviations from this. This allows for non-normally distributed data to be effectively analysed without the necessity to ―pre-whiten" data or use one-way statistical transforms on the data.

The paper is organised as follows. Section 1 has introduced the motivation for this research, with Section 2 discussing the related literature. The dataset employed is described in Section 3. Following this, the analytical model is presented in Section 4, with experimental design in Section 5. Results are presented in Section 6 with discussions and conclusions following in Section 7and 8 respectively.

2. RELATED WORK

As previously stated, data-mining techniques are often ineffective in practice due to the large bias in favour of the majority class – typically normal operational behaviour – which reduces the incentive for machine learning algorithms to truly encapsulate failure behaviour. This occurs as in a dataset with 0.1% failure data, the system can achieve a classification accuracy of 99.9% by merely returning the default case (Godwin & Matthews, 2014).

Many algorithms have been proposed to remove the inherent bias in unbalanced datasets (such as in the realm of prognosis). These fall into two main categories, namely under-sampling and over-sampling. Under-sampling removes data from the majority class to remove the bias, whereas over-sampling adds data to the minority class. As such, these techniques will often either reduce the information content in the data, or create synthetic data which needs to be validated and verified. For a full review of data balancing techniques, please refer to Baydar et al., 2001.

It should be noted that these techniques often require labelled data (Baydar et at., 2001). In practice, this is often not available (as failures are yet to occur), or it is too costly to manually label high frequency data. As such, analysis of high frequency data should be performed by statistical techniques which can exploit the high frequency nature of the data to increase the statistical power of the results.

High frequency data is often employed for bearing prognosis due to the ability to extract time, time-frequency and frequency domain features. This enables the use of many different techniques to assist in the diagnostic and prognostic process.

Amongst the most commonly used techniques for bearing diagnosis and prognosis is that of the fast Fourier transform (FFT) (Rai & Mohanty, 2007). This is a frequency domain signal that can be used to detect degradation and identify failure modes. Work done by (Zappalà et al., 2013) uses sideband analysis of key harmonic frequencies in order to monitor the degradation of components over time. As sideband analysis utilises specific harmonic frequencies, the relationship between the harmonic and the immediate sideband frequencies can be analysed as degradation occurs. As such, the technique can be applied where traditional frequency domain techniques are not as powerful (such as in non-stationary signal analysis), for instance, in wind turbine gearbox analysis (Zappalà et al., 2012).

Various other techniques for frequency domain analysis have been explored for rotating machinery such as gearboxes and bearings. Typically, these involve the use of the power spectrum (Ho & Randall, 2000) or Cepstrum analysis (van der Merwe & Hoffman, 2002).

The most commonly utilised domain for frequency analysis is that of the time-frequency domain. Within this, the use of the wavelet transform (Raffiee et al., 2010) is prevalent. Due to the ability to combine frequency domain information in conjunction with time domain data (Raffiee et al., 2010), many strong prognostic signatures can be identified in these techniques.

The wavelet transform is employed due to its ability to remove noise from the data. As various wavelet functions exist (known as mother wavelets), different signatures and artefacts from high frequency data can be discovered and used for diagnostic and prognostic analysis (Lin & Zuo (2003), Peng & Chu (2004), Jardine et al., 2006).

Recently, the use of time synchronous averaging (TSA) has become more prevalent in the literature for prognosis of

high frequency data such as bearings and gearboxes (Bechhoefer et al., 2013). This technique is a hybrid time-frequency technique which employs a tachometer in order to deduce the current orientation of the rotating component. This enables further information to be gathered in the prognostic process, such as the identification of specific bearing roller elements which have degraded or if a specific gear tooth has degradation. Derivations of TSA exist which do not require a tachometer (Bechhoefer et al., 2009); however, these often simply estimate the tachometer signal. For a review of TSA techniques as applied to health assessment, please refer to the extensive review undertaken by (Bechhoefer et al., 2009).

Within the time-domain, often statistical features are extracted from the signal. Commonly in the literature, skewness and kurtosis are employed for diagnosis and prognosis (Heng & Nor, 1998 and Tandon, 1994). Skewness is the third standardised moment and represents the asymmetry of an underlying distribution, whereas Kurtosis is the fourth standardised moment and represents the peaked-ness of the underlying distribution.

In practice, due to the high frequency of the data, it is often assumed that the data is normally distributed due to the central limit theorem. As the behaviour of the normal distribution is well understood, we can exploit a-priori knowledge for prognosis. Typically, for a healthy bearing or gear, little to no skewness will exist in the data, and the peaked-ness of the data will typically be 3. However, these features are not reliable for a variety of reasons. When used in uni-variate models, it is possible for the underlying distribution of the data to change due to factors such as degradation, without effecting the skewness and kurtosis of the distribution. As such, the use of these features without additional context (additional features, a-priori knowledge or otherwise) should be avoided.

It should also be noted that typically accelerometer data is employed for analysis in all three commonly used domains. However, the use of acoustic emission (AE) sensor data is becoming more widespread due to potentially increased sensitivity (Bechhoefer et al., 2009) in a variety of methods.

Other time domain features can be used for diagnosis and prognosis. Amongst the most reliable time domain feature is that of oil analysis through the use of oil debris monitoring systems (Feng et al., 2012). These systems are able to monitor the particulate level in parts per million (PPM) in the oil of an asset in order to infer information regarding degradation or potential future failure modes (Feng et al., 2012). These systems are used extensively within the wind industry for monitoring of the gearbox, which is of critical importance (Stephens, 1974). However, these sensors are currently prohibitively expensive for practical use in non-mission-critical scenarios.

As the use of skewness and kurtosis requires making assumptions regarding the underlying distribution of the data, and may not accurately reflect the true change in condition, new techniques are needed. A robust uni-variate nonparametric approach to mitigate these issues can be derived by employing empirical statistical techniques. To demonstrate this, publically available data is employed.

3. DATASET DESCRIPTION

For the following series of experiments, publically available data was employed for transparency. The data was collected by the centre for intelligent maintenance systems (IMS), with the support of the Rexnord Corporation, and made available by NASA (Lee et al., 2007).

Four bearings (force lubricated) were installed onto a shaft which was kept at a constant 2000 RPM by an AC motor. A 6000 lbs radial load was applied via a spring mechanism to the shaft. Rexnord ZA-2115 double row bearings were used, with data collection performed by a National Instruments DAQ 6062E. The accelerometers used in the experiment were PCB 353B33 High Sensitivity Quartz ICP accelerometers. Data was sampled at 20 KHz, equating to 20,480 samples per second. Data was sampled every 10 minutes until oil debris monitoring equipment reached a particulate count which indicated bearing failure. At this point the data collection was deemed complete, and the bearings were removed for inspection. All bearings exceeded their design life expectation. Vibration data pertaining to acceleration was collected during rotational operation, and is measured in G.

4. MODEL DEVELOPMENT

Due to the cases which exist when employing skewness or kurtosis in time series analysis for prognosis, new prognostic features must be developed. In order to ensure that new features do not suffer from the same pitfalls of skewness and kurtosis, 3 factors must be taken into consideration.

Firstly, the technique should be nonparametric. As such, little to no assumptions regarding the underlying data is required. This would enable the technique to work as effectively on normally distributed data as data which is not ordinarily normally distributed, as is often the case in practice for prognostic applications. Secondly, the technique should be robust to noise. Noise is inherent in all real-world signals, and as such, techniques should be robust to this. By identifying data which may potentially be anomalous, this can be disregarded or exploited for further prognosis.

Finally, the technique should accurately respond to changes in the condition of the asset. Skewness and kurtosis have the potential to remain constant whilst degradation occurs. Whilst this may seem trivial, cases such as this should

always be checked to ensure that degradation is always observed.

As such, in this work, we propose the use of the two-sample Kolmogorov-Smirnov test (Stephens, 1974) for the diagnosis and prognosis of bearing condition. This is a non-parametric uni-variate technique which can be employed to compare a sample with a given distribution to quantify and signify significant deviations.

The two-sample test statistic quantifies the distance between two cumulative density functions (empirically derived or otherwise). This enables the test statistic to be used as a prognostic health index by fixing one sample to a known state of normal operation behaviour. Thus, it is expected that should degradation occur the distribution of the underlying data will change accordingly. Differing levels of statistical significance can be employed to identify inspection, maintenance and replacement thresholds, with a prognostic time series derived by plotting the changes of the statistic over time.

The Kolmogorov-Smirnov test can be defined as follows (Stephens, 1974):

$$D_{n,n'} = \sup_x |F_{1,n}(x) - F_{2,n'}(x)| \quad (1)$$

Where $\sup_x$ refers to the supremum of set x, and $F_{1,n}$ and $F_{2,n'}$ refer to the empirical distribution function, defined as:

$$\mathrm{F(x)} = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \leq x} \quad (2)$$

Where *I* refers to the indicator function, defined as:

$$I_{\mathrm{x_i \leq x}} = \begin{cases} 1 \; if \; X_i \leq x \\ 0 \; otherwise \end{cases} \quad (3)$$

As such, the test statistic D (as in Eq. 1) represents the maximum difference between the empirically defined distribution F_1 and F_2.

Thus, for a given behaviour, it is possible to accurately measure the deviation from this behaviour and determine its statistical significance. This enables the creation of a health metric as described in the following Section.

5. EXPERIMENTAL SETUP

In order to determine deviations from a known state, a-priori knowledge of the know state must be utilised within the model. Previous work which utilises the Kolmogorov-Smirnov test pre-whitens the data (Cong et al, 2011). Pre-whitening of the data ensures that the data is effectively white noise mixed with the transient signal of the bearing. As such, it is possible to employ a one sample Kolmogorov-Smirnov test for the purposes of bearing degradation assessment by sampling against a Gaussian distribution.

Whilst this removes the need for a-priori knowledge as the effective sample from which degradation is measured, it also infers assumptions regarding the underlying data.

For instance, with regards to the NASA bearing dataset, normality testing was performed via the highly sensitive Anderson-Darling test (Anderson & Darling, 1954). This is a one sample non-parametric test with higher power than the Kolmogorov-Smirnov test, and is computed by:

$$A = -n - \frac{1}{n}\sum_{i=1}^{n}[2i-1][\ln(p_{(i)}) + \ln(1 - p_{(n-1+i)}] \quad (4)$$

Where $p_{(i)} = \Phi([x_i - \bar{x})]/s)$ where Φ refers to the CDF of the normal distribution, and $\bar{x}, s$ refer to the mean and standard deviation of the data (respectively).

Within the 2nd set of NASA bearing data, 4 bearings across 984 files were assessed for normality. Of the 3936 normality assessments, 16 samples ($< 0.5\%$) of the bearing data were normally distributed ($p < .05$). As such, given the large sample size (20,480) of each sample, we can infer that the underlying structure of the data is not normal. This is expected; however, as previous work pre-whitens the data, it may be the case that pre-whitening of the data synthetically manipulates the data to ensure normality. Whilst this is effective, it is also computationally intensive, and has the ability to swamp or mask the true bearing signal (Bendre, 1989) and increase noise within the signal.

By replacing the normal distribution reference sample with a known behaviour, we remove the computational intensity, reduce the number of assumptions regarding the underlying data and also reduce the noise within the signal.

In order to explore the use of the Kolmogorov-Smirnov test for the diagnosis and prognosis of bearing faults, three experiments were performed, with an additional experiment utilising the one sample Anderson-Darling test for comparison.

In the first experiment, the Anderson-Darling test is used to quantify the deviation of the data from the normal distribution. This experiment explores the relationship between the normal distribution and the degradation of the bearing. It is expected that as the bearing degrades, the deviation will increase, and can be used to quantify the current level of degradation on the bearing. The second experiment employs the Kolmogorov-Smirnov test without the use of a-priori knowledge. In this case, each data sample is tested against the previous sample to quantify the degradation which has occurred in the previous 10 minutes. Significant degradation of the bearing which occurs between samples are expected to be revealed by this test. The third experiment employs a-priori knowledge to fix a sample point from normal behaviour within a bearing, from which all samples are then measured against. Although this requires the use of a-priori knowledge (in the form of normal operational behaviour), the authors believe this trade off is practical due to normal operational behaviour relating to the majority class. In order to validate the approach, in this experiment, data from a single bearing is employed (2nd

test, bearing 1). As this bearing is known to fail, this experiment is intended to prove the Kolmogorov-Smirnov test as a viable time domain feature for diagnosis and prognosis. In the final experiment, data from a healthy bearing is employed as the sample for the Kolmogorov-Smirnov test. This mitigates the practical issues which occur in the third experiment (namely, use of data sampled from a bearing which failed which may not be available in practice) to increase the viability of the approach. As many bearings are subjected to identical conditions (for instance, in a production facility or wind turbine), by utilising known normal behaviour of a single bearing, the approach can systematically be applied to all of the assets in the facility individually.

6. RESULTS

In the first experiment, the Anderson-Darling test is employed as a non-parametric one sample statistical test to measure deviation from the normal distribution. As degradation is expected to cause deviations from this distribution in mean value, standard deviation, skewness, and kurtosis, this test should perform well. However, as can be seen in Figure 1, this is not the case.

Figure 1 (a) presents a healthy bearing and a failed bearing over time (Bearings 1 & 2 from the 2nd set of test data (Lee et al., 2007)) as measured by the p-value of the Anderson-Darling test statistic. Although the healthy bearing line remains stable, the test only identifies a single peak on the failed bearing. Although this is over 46 hours before failure, no progressive trend is observed. As degradation is often an exponential phenomenon, the log plot of Figure 1 (a) is taken and presented in Figure 1(b). This is the natural transformation of exponential data. Although degradation phenomena is observed much earlier due to this transformation (at over 67 hours before failure), there are many inconsistencies with the trend; for instance, degradation seems to decrease and increase over many cycles. Although this does provide insight into the underlying characteristics of the bearing, it violates the prognostic principles metrics must adhere to set out in section 4. The second experiment employs the two-sample non-parametric uni-variate Kolmogorov-Smirnov test to quantify degradation based upon the empirical CDF of the data. Each data sample is compared to the previous collected data sample to determine significance which may imply degradation has occurred.

Figure 2 presents the Kolmogorov-Smirnov D statistic for both the same healthy and failed bearing as in the previous experiment. As can be seen in Figure 2(a), both time series appear to be highly correlated. A Pearson product-moment correlation coefficient was computed to assess the relationship between the healthy bearing, and the failed bearing, and were found to be highly correlated ($r = .97$). It is interesting to note that the peak which has been highlighted in Figure 2(a) is identified in both bearings, and may be due to external factors which occurred during the data collection process. Figure 2(b) presents the log-transform of Figure 2(a). Again, it is difficult to separate the healthy bearing from the failed bearing as no obvious signatures are apparent. Figure 2(c) shows the p-value of the Kolmogorov-Smirnov test for each bearing. It can be seen that this is limited in its use for diagnosis and prognosis, due to many false positives in early life and many false negatives when degradation has occurred. The third experiment exploits these results by fixing the sample to a constant behaviour, from which deviations are then computed. Although this requires a-priori knowledge, this can be taken from OEM documentation. As in this case, it is essential that the fixed points contain no degraded behaviour, the point from which the sample is fixed directly correlates to the quality of the metric which is derived. As such, we exploit historical data in conjunction with OEM documentation and traditional reliability analysis to determine normal behaviour. As each bearing has a design life of 1 million revolutions and the experimental setup ran the bearings at 2000 RPM, we can easily determine from the time elapsed, a percentage of expected useful life. Due to the existence of infant mortality due to manufacturing defects as commonly presented by the so-called ~~"b~~athtub curve" (Leemis , 1995) we can then define a point or a set of points which are likely to correspond to normal operational behaviour. For simplicity, data taken from 10-15% of asset life was utilised in this experiment. The first 10% of asset life is not taken into consideration due to the possibility of manufacturing defects or potential infant mortality.

Figure 3 shows the same healthy bearing and same failed bearing when a fixed sample is chosen for the two-sample Kolmogorov-Smirnov test. In practice, we would not retrospectively analyse the first 15% of bearing life, however, for completeness, this has been left in Figure 3. As can be seen in Figure 3(a), for the failed bearing, a strong prognostic signature is detected when employing the D statistic from the Kolmogorov-Smirnov test. Exponential degradation is present, and can be identified as early as 75 hours prior to failure. Initially, a linear trend is found to occur, this is followed by healing phenomena, which afterwards reverts to exponential degradation. Figure 3(b) depicts the logarithmic transform of same experiment, with the artefacts mentioned above highlighted. It should be noted that the same artefacts as in experiment two are observed at the beginning of the time series, which is of interest. The healthy bearing is found to be consistently healthier than the failed bearing, which is promising. Similarly, the D-value remains stable during operation, with exponential degradation occurring at the end of life. This shows the potential of the Kolmogorov-Smirnov test as a prognostic index for bearing health assessment.

The D statistic is employed due to its many features which are complementary for reliability engineering analysis, and

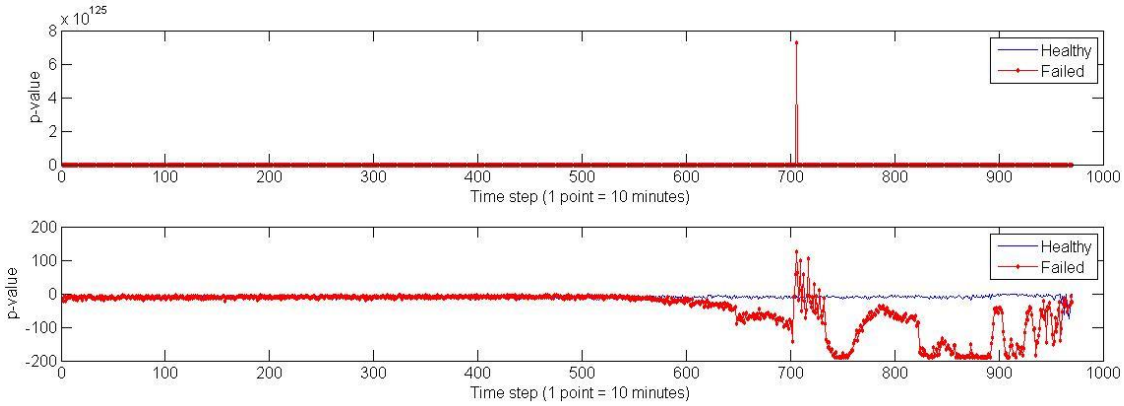


Figure 1. Anderson-Darling test for degradation, showing (a - top) raw values, and (b - below) the logarithmic transform.

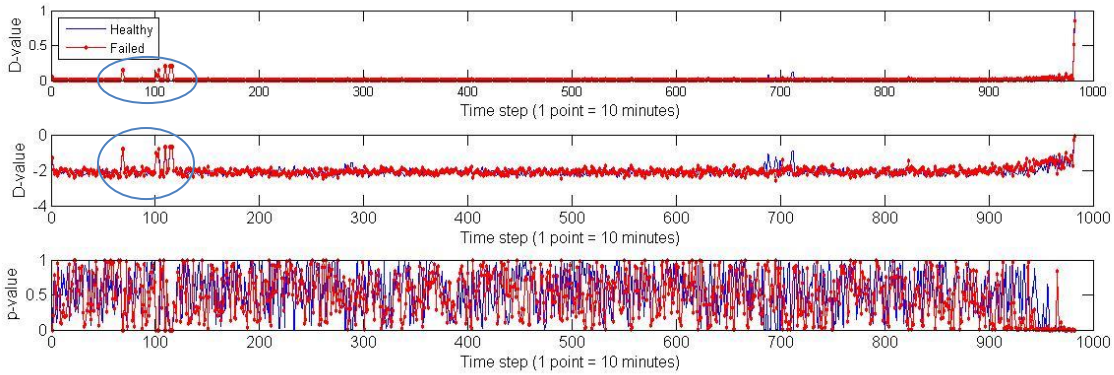


Figure 2. Two sample, transition based, Kolmogorov-Smirrnov showing (a - top) raw D-statistic, (b - centre) the logarithmic transform and (c - below) the associated significance (p-value).

prognostics in general. For instance, the D statistic is bounded between 0 (no difference in the distributions) and 1 (maximum difference in the distributions). As such, it is expected to increases as degradation occurs (as in Figure 3). This bounding also provides a simple means to estimate the percentage of useful life used.

Figure 3(b) shows the log-transform of Figure (A). This then presents the degradation which occurs as a linear phenomenon. This then enables further statistical analysis, such as regression analysis to perform remaining useful life (RUL) estimation for some given condition (D-value). In addition to the D-value being employed, the p-value of the test allows a natural extension of this analysis. If we are to check significant deviations ($p < .05$), the first consistent (repeated 3 times or more) significance is found 73 hours prior to failure, and remains significant until failure (on the failed bearing). For the healthy bearing, consistent significant deviations are found 17 hours prior to the end of the test, which may refer to the initial stages of degradation on the bearing. As such, the use of various p-values can be seen as an effective means for identifying inspection of maintenance activities for decision making within enterprise.

In the final experiment, the fixed sample in the Kolmogorov-Smirnov test was derived as in the previous experiment, however, from an independent bearing which did not fail (Bearing 3, test 2 (Lee et al., 2007)). This experiment explores the versatility and generalisability of the technique. If the bearings are subjected to similar conditions, then normal behaviour of each bearing should be similar. As such, regardless of the bearing used to fix the first sample, the deviation from this should correlate highly to the results achieved in experiment 3. Figure 4 shows the healthy bearing and failed bearing when the fixed sample used for the analysis is from an independent bearing. As expected, this is similar to the results achieved in experiment 3. A Pearson product-moment correlation coefficient was computed to assess the relationship between the D-statistic of the failed bearing taken from experiment 3, and the D-value taken from the failed bearing in experiment 4. These were found to be highly correlated ($r = .86$). Similarly, a further Pearson product-moment correlation coefficient was computed to assess the same relationship for the healthy bearing. This was again found to be highly correlated ($r = .97$). This shows the effectiveness of the technique when applied to new bearings which are expected to operate in similar

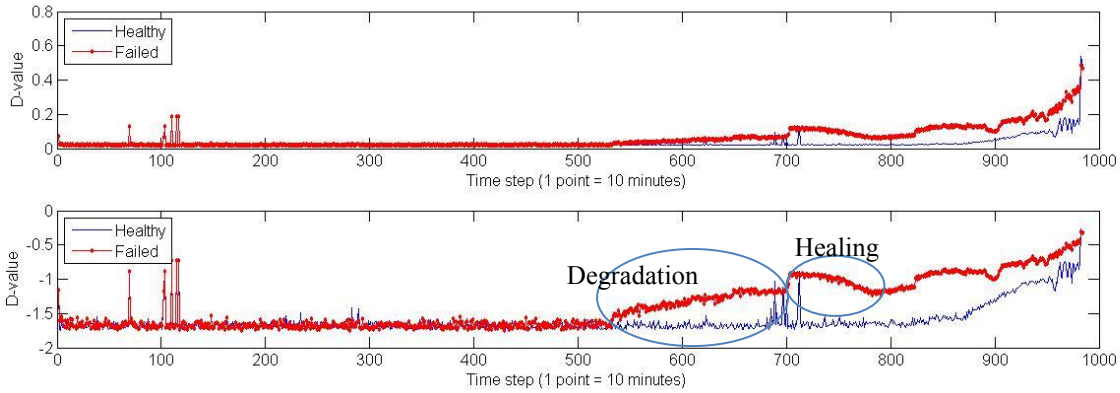


Figure 3. Two-sample, fixed Kolmogorov-Smirnov test, showing (a - top) raw *D*-statistic and (b - below) the log transform.

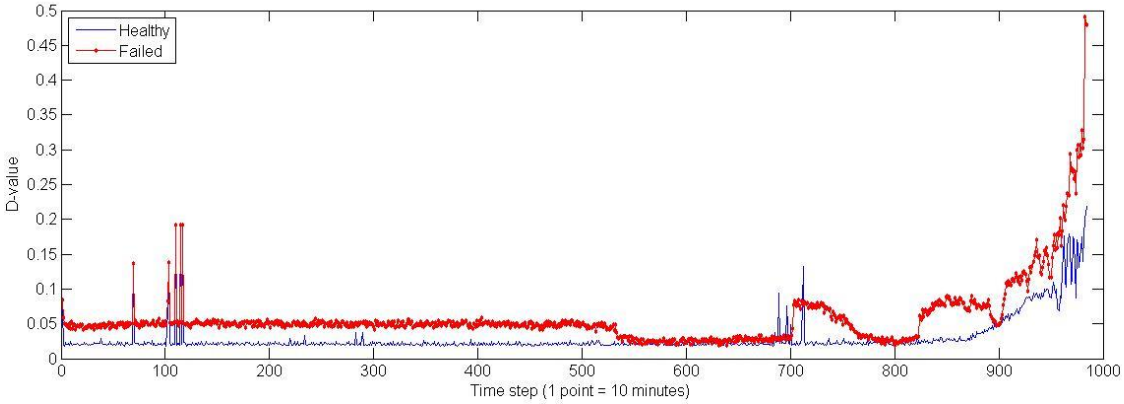


Figure 4. Independent verification of experiment 3 (Figure 3(a)) showing raw *D*-value.

conditions to those which the fixed sample was derived from.

With regards to the significance of the p-values derived from the final experiment in relation to the prognostic horizon, the sensitivity of the technique hinders the benefit gained. As in this case, a 6000 lbs radial load was applied to the shaft, this affects each bearing in a different way. As such, the underlying distributions are inherently different, and thus differ significantly. This then makes each observation appear to be significantly different. However, it is still possible to use the degree of significance as a means for prognosis, as the p-value continues to decrease in proportion to the degradation apparent in the bearing.

7. DISCUSSION

In the first experiment, the Anderson-Darling test was used as a one-sample test in order to mitigate the necessity of a-priori knowledge. However, in this case, the data is not normally distributed and as such, this technique is not effective. In other systems where high frequency data is normally distributed, this may be more sensitive than the Kolmogorov-Smirnov test, and as such, should be used initially.

The Anderson-Darling test is used in the initial analysis over the Shapiro-Wilk test due to the high frequency nature of the data involved. The Shapiro-Wilk test is highly sensitive for large sample sizes, and as such, rejects the null hypothesis often.

As both the Anderson-Darling and Shapiro-Wilk tests are one-sample, they cannot be utilised to empirically derive the CDF of the underlying data, and as such, if the data is not normally distributed, cannot be used to identify deviations specifically from the distribution of the data in question.

It is interesting to note that the artefacts at the start of the time series which can be observed in figures 2 through 4 do not occur in figure 1. This is likely due to the insensitivity of this test due to the underlying distribution of the data. The cause of these artefacts is currently unknown; as similar artefacts are observed throughout both bearings it has been inferred that this is due to the experimental setup and external factors associated with this. The artefacts in figures 2 through 4 for the healthy bearing at approximately time step 700 are unexplained. This could potentially be due to the development of degradation on the failed bearing (from time step 550 as per figure 3) causing particulates in the oil

which were transferred to this bearing and ultimately resulted in degradation on the healthy bearing.

The reduction in D-value observed in figure 4 should also be noted. This is an artefact caused by employing a different bearing (with slightly different manufacturer tolerances and defects) in a different bearing position in the experimental setup as a reference. This was undertaken as a proof of concept and I practice, as each bearing will behave in a unique way, historical data pertaining to the bearing in question should be employed.

With regards to fixing the data representing normal behaviour for the two-sample Kolmogorov-Smirnov test, it is essential that no degradation is incorporated into this sample. This is difficult to determine a-priori.

One solution to this would be to use robust outlier analytical techniques to derive a sound subset across the full life of one bearing. As the operational behaviour of the bearing would dictate degradation to be outlying, this would effectively be removed.

In practice, the use accelerometer data is not ideal for robust analysis due to the limited sensitivity of the data collection equipment. If robust techniques such as Median Absolute Deviation (MAD) are used to remove outliers, significant parts of the distribution tails are removed. This limits the effectiveness of the two-sample Kolmogorov-Smirnov test due to the resultant effect on the empirical CDF, which inherently increases the noise within the derived prognostic. The authors recommend not using robust outlier removal in conjunction with accelerometer data, as by their definition, outliers are inherently beneficial for prognosis.

In the case where acoustic emissions (AE) sensors are employed, due to increased sensitivity, the use of robust outlier techniques can potentially be employed effectively.

8. CONCLUSION

This paper has shown the viability of the use the two-sample uni-variate Kolmogorov-Smirnov test as a means to derive low-frequency time-domain prognostic signatures from high frequency data. The versatility of the technique is explored with publically available data (Lee et al., 2007).

Strong prognostic signatures are found for both bearings on which analysis was performed as early as 54.2% of the bearings life (for the failed bearing), and 89.6% of bearing life (for a bearing which ultimately did not fail).

By empirically deriving the CDF function of the data, external conditions are inherently considered and taken into account by the prognostic system. Although this requires a-priori knowledge (historical high frequency data), should this not be available, the empirical function could be approximated by establishing the underlying distribution and using the exact CDF of the chosen distribution.

Although the technique is versatile, it cannot be applied to non-stationary techniques; the transient nature of the signal would almost certainly ensure that statistically significant deviations from the pre-defined normal behaviour are consistently observed whilst no degradation is present: this would violate the prognostic principles laid out previously. For the purposes of this work stationary is defined as a lack of temporal dependency of the marginal distribution (i.e., the distribution of the bearing values does not change with time).

Future work will look to extend this analysis to non-stationary signals for wind turbine gearbox analysis by normalising for loading transitions. The signal can be broken into a series of stationary signals with transient periods which can be identified by correlating the data with the onboard SCADA system.

ACKNOWLEDGEMENTS

This research is funded through an EPSRC Industrial CASE award in collaboration with 5G Technology Ltd

BIOGRAPHIES

Jamie L. Godwin is working towards the degree of Doctor of Philosophy within the department of Engineering and Computing sciences at the University of Durham. He has researched areas such as SCADA data analysis and robust multivariate prognostic techniques. His current doctoral research focuses on metrics for maintenance effectiveness, SCADA data analysis and robust multivariate statistical measures for prognosis.

Peter Matthews is a Lecturer in Design Informatics at the School of Engineering and Computing Sciences. He is the author or co-author of numerous books, technical papers and EU patents. His current research interests are centred around industrial data analysis involving collecting and analysing data obtained either from production process monitoring (e.g., SCADA logs) or service life data (e.g. maintenance logs) and utilising Monte Carlo simulations, Evolutionary Algorithms, Bayesian Belief Networks to form future design and operation decisions.

REFERENCES

Verma, A. and Kusiak A., (2011). *Predictive Analysis of Wind Turbine Faults: A Data Mining Approach*. in Proceedings of the 2011 Industrial Engineering Research Conference, Reno, Nevada, May 19-23,1 – 9.

Levrat, E., Iung, B., Crespo, (2008). E-*maintenance:f review and conceptual framework*. Production Planning & Control 19 (4), 408429.

Hameed, Z., Hong, Y., Cho, Y., Ahn S., and Song, C., (2009). *Condition monitoring and fault detection of wind*

turbines and related algorithms: A review. Renewable and Sustainable energy reviews, 13(1), 1–39.

Lee, J., Qiu, H., Yu, G., Lin, J., and Rexnord Technical Services (2007). *'Bearing Data Set'*, IMS, University of Cincinnati. NASA Ames Prognostics Data Repository, [http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/], NASA Ames, Moffett Field, CA.

Godwin, J. L., & Matthews, P. C., (2014a) *Robust Statistical Methods for Rapid Data Labelling.* In V. Bhatnagar, "Data Mining and Analysis in the Engineering Field". IGI Global : Hershey, PA, USA.

Baydar, N., Chen, Q., Ball, A. and Kruger, U. (2001). *Detection of incipient tooth defect in helical gears using multivariate statistics*. Mechanical Systems and Signal Processing, 15 (2), 303—321.

Rai, V. K., & Mohanty, A. R. (2007). *Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform*. Mechanical Systems and Signal Processing, 21(6), 2607-2615.

Zappalà, D., Tavner, P., Crabtree, C., & Sheng, S. (2013, January). *Sideband Algorithm for Automatic Wind Turbine Gearbox Fault Detection and Diagnosis*. In European Wind Energy Association (EWEA) Conference, Vienna, Austria.

Zappalà, D., Tavner, P. J., & Crabtree, C. J. (2012). *Gear fault detection automation using WindCon frequency tracking*. In Proceedings European Wind Energy Conference.

Ho, D., Randall, R.B., (2000) *Optimisation of bearing diagnostic techniques using simulated and actual bearing fault signals*, Mechanical Systems and Signal Processing 14 (5) pp. 763–788

N.T. van der Merwe, A.J. Hoffman, (2002). *A modified cepstrum analysis applied to vibrational signals*, in: Proceedings of 14th International Conference on Digital Signal Processing (DSP2002), vol. 2, Santorini, Greece, 2002, pp. 873–876

Raffiee, J., Raffiee, M., Tse, P., 2010. *Application of mother wavelet functions for automatic gear and bearing fault diagnosis*. Expert Systems with Applications 37 (6), 4568 - 4579.

Peng, Z., Chu, F., (2004). *Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography.* Mechanical Systems and Signal Processing 18 (2), 199 - 221.

Lin, Zuo, M., (2003). *Gearbox fault diagnosis using adaptive wavelet filter*. Mechanical Systems and Signal Processing 17 (6), 12591269.

Jardine, A. K., Lin, D., Banjevic, D., (2006). *A review on machinery diagnostics and prognostics implementing condition-based maintenance.* Mechanical Systems and Signal Processing 20 (7), 1483 - 1510.

Bechhoefer, E., Qu, Y., Zhu, J. and He, D. (2013) *Signal Processing Techniques to Improve an Acoustic Emissions Sensor*. In proceedings of the annual conference of the PHMsociety October 14 – 17, New Orleans, LA, USA. 4(3). pp. 18.

Bechhoefer, E., & Kingsley, M. (2009). *A review of time synchronous average algorithms*. In Annual conference of the prognostics and health management society.

Heng, R.B.W., Nor. M.J.M., (1998) *Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition.* Applied Acoustics, 53 (1-3), pp. 211–226

Tandon, N.. (1994) *Comparison of some vibration parameters for the condition monitoring of rolling element bearings*. Measurement: Journal of the International Measurement Confederation, 12 (3), pp. 285–289

Feng, Y., Qiu, Y., Crabtree, C., Long, H. & Tavner P., (2012) *Monitoring wind turbine gearboxes*. Wind Energy. Vol 16(5). pp. 728 – 740.

Crabtree, C.J. (2010). *Survey of Commercially Available Condition Monitoring Systems for Wind Turbines*, SuperGen Wind.

Stephens, M. A. (1974). *EDF Statistics for Goodness of Fit and Some Comparisons*. Journal of the American Statistical Association (American Statistical Association) 69 (347): 730–737

Cong, F., Chen, J., & Pan, Y. (2011). *Kolmogorov-Smirnov test for rolling bearing performance degradation assessment and prognosis*. Journal of Vibration and Control, 17(9), 1337-1347.

Anderson, T. W., & Darling, D. A. (1954). *A test of goodness of fit*. Journal of the American Statistical Association, 49(268), 765-769.

Bendre, S. M. (1989). *Masking and swamping effects on tests for multiple outliers in normal sample*. Communications in Statistics-Theory and Methods, 18(2), 697-710.

Leemis, L. M. (1995). *Reliability: probabilistic models and statistical methods*. Prentice-Hall, Inc..

Health Management and Diagnostics for Synthetic Aperture Radar (SAR) Payloads

Gregory Bower[1], Jonathan Zook[1] and Ross Bird[1]

[1]*QorTek Inc., Williamsport, PA, 17701*

gbower@qortek.com
jzook@qortek.com
rbird@qortek.com

ABSTRACT

A statistical method based on symbolic analysis is presented for health management of Synthetic Aperture Radar systems. The approach, based on symbolic theory, develops statistical models of the underlying system dynamics using an underlying Markov assumption and tracks the change in model over time to determine system health. The methodology was designed for minimal impact to legacy systems and required minimal computational effort in order to operate at radar data rates. The approach was applied to radar phase history data corrupted with simulated degradation. Two degradation mechanisms were studied: interference and array degradation. In addition, the results of combined degradation were also studied in this work.

1. INTRODUCTION

Health management of systems can result in the reduction of necessary man-hours and costs associated with maintenance of equipment. In addition, a health management routine can be used to determine the remaining useful life of a system and to determine when to schedule upcoming repairs. Data driven methods utilize data captured in real-time from the system in order to determine the current state of health of the system. Data driven methods form underlying models of the system using this captured time series data. These underlying models developed through operation of the system can then be used to quantify remaining health.

The method was originally applied to monitoring the health of a dc-dc forward converter in order to predict the remaining useful life of the converter (Bower, Mayer, & Reichard, 2011)(Bower, Mayer, Reichard, 2008). The Markov assumption is implied for the system under investigation from which statistical models are developed and tracked through time. Increasing degradation results in perturbing the operational characteristics of the system which can result in a shift in the Markov process (Papoulis & Pillai, 2002). This shift can be quantifiable and with proper training, predictable in the future for prognostic purposes.

In this work, a symbolic approach was adopted for health monitoring of imaging radar payloads on Unmanned Aerial Vehicles (UAVs). These radar platforms are complex systems difficult to model classically which makes the proposed data based approach ideal for health monitoring. The primary objective of this research was to determine the feasibility of applying such a method to the high data rates seen in an imaging radar platform which is a product of the pulse repetition rate of the radar at the desired sample rate and bandwidth of the return echoes. In addition, the approach cannot interfere with the operation of the platform or radar system. The methodology was tested with radar phase history data and two common issues with imaging radars, interference and array degradation were investigated. The results are also expected to lead to an ability to discriminate between the two degradation mechanisms to assist in optimizing the operation of the radar payload. This paper begins with a discussion on the Symbolic Analysis approach specifically applied to the imaging radar payload and all details of the approach are discussed. In Section III, a brief review of Synthetic Aperture Radar and radar platforms is completed. Section IV reviews the results obtained from the simulations and feasibility testing of the approach and the paper concludes with future work in Section V.

2. SYMBOLIC ANALYSIS

Symbolic Analysis is a statistical pattern recognition tool based upon symbolic theory. Most work in the symbolic realm deals with the development of optimal models to determine the trajectory of modeled system states (Daw, Finney & Tracy, 2003). These methods are used to model complex and chaotic systems. The resultant optimal model, known as the ε machine, has a variable dimensional structure whose dimensions were constantly adjusted depending on the data collected over time. This variation in dimensionality made it difficult to determine deviations

Gregory Bower, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. Approved for public release: 88ABW-2013-3610.

between models developed through system usage. In order to make meaningful comparisons between models, a machine was developed with a-priori fixed dimensional structure (Ray, 2004). This fixed dimensional machine allows for meaningful comparisons between statistical models defined at different temporal points in the system's life at the cost of optimality. Using the SA approach, it is possible to generate a measure that quantifies the amount of degradation within a recorded observable. The process of SA is shown in the block diagram of Figure 1. The basic methodology requires four steps which will be detailed in the next sections.

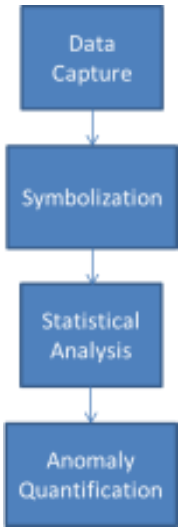


Figure 1. Symbolic analysis of time series data block diagram.

2.1. **Data Capture**

Although the process of data capture might seem straightforward, the process requires some careful consideration. First, the type of data and where it is captured must be known. This entails the study of the underlying system in order to determine the common failure points of the system. Once these failure points are known, the rate and length of the data to be recorded must be determined.

Symbolic analysis requires two assumptions. First, it was assumed that the degradation within the system monotonically increases. This means that the system does not undergo 'self-healing' or is repaired during the monitoring process. Limiting self-healing is important for the implementation of remaining health estimation. Secondly, it was assumed that the degradation mechanisms act slower than the system dynamics. This assumptions states that when the system is observed and the time series data collected, that the degradation in the system during this period was assumed to be constant. In this manner, a model of the system was developed based on the constant state of degradation.

For the application to radar platforms, specifically SAR systems, the data implemented in the algorithm was the fast time scale which was developed from an individual pulse (phase history data). The slow time scale was defined to be the pulse rate or repetition rate of the platform.

2.2. **Symbolization**

The next step involves transforming the time series data into the symbolic domain. This step can be thought of as a general re-quantization of the original data resulting in a coarser distribution. Symbolization requires the determination of the number of partitions to be used as well as the type of partitioning. The two most common types of partitioning include uniform partitioning (UP) and maximum entropy (ME) partitioning. The choice in the number of partitions will depend on the time series data being analyzed as well as the type of degradation and features to be analyzed.

The partitioning was kept invariant over the entire monitoring period such that the statistical models developed later in the system life can be directly compared to the baseline. The baseline model was defined on the healthy state of the system.

2.3. **Uniform Partitioning**

Uniform partitioning divides the range of the time series data into equal sized regions where the total number of determined partitions are defined as the set P. Given the range of the time series data as U, the partition sizes are defined as U/P and the boundaries developed from the range U. Each partition region P_i was mutually exclusive and exhaustive over the range of the data. The probabilities of the partition occurrence in the uniform case are not necessarily equal; however, the partitioning structure was equal.

To construct UP, the maximum and minimum of the time series data were evaluated and the resultant range was divided equally into P regions. These regions are assigned a unique symbol to complete the partition description. An example of UP on a sinusoidal waveform is shown in Figure 2.

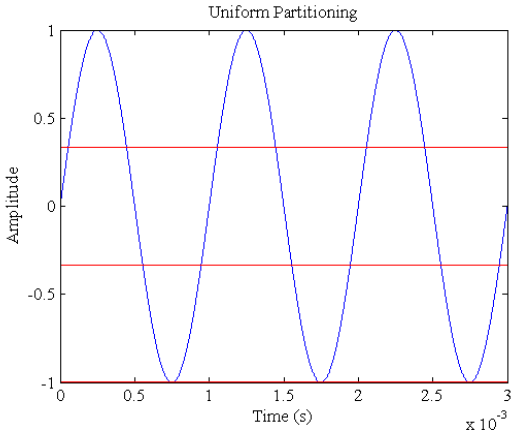


Figure 2. Example of uniform partitioning of a sinusoid.

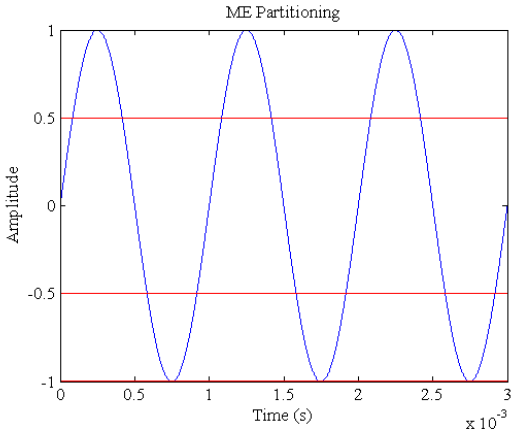


Figure 3. Example of ME partitioning of a sinusoid.

2.3.1. **ME Partitioning**

The maximum entropy (ME) partitioning scheme was defined by the principle of entropy in determining the partition structures. Recall entropy as shown in Eq. 1.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \qquad (1)$$

The entropy can be maximized by setting $p(x_i) = p(x_j), \forall i, j$. The logarithm to base 2 was used so that the unit of entropy is in bits. In the time series data, accomplishing maximization of entropy in the baseline case was necessary to make sure all partitions (or symbols) have equal probability of occurrence. The partition structure resulting from ME does not necessitate equal partitions as in the uniform case but does guarantee equal prior probabilities for the partitions in the baseline case. A feature of the ME partitioning scheme is that the partitions boundaries are closer in regions of the data where there are a dense number of data points. In regions where there are fewer date points, fewer partitions are generated in these areas. An example of ME partitioning on a sinusoidal signal is shown in Figure 3. For the ME case, the resultant probability of the symbols was equal compared to uniform partitioning whereas the partition regions are equal in size with unequal symbol probabilities.

Once the partitions are defined each partition was labeled with a symbol from the alphabet *S*. Given a time series *X* of length *M*, if $x_i \in P_i, 0 \leq i \leq M$, then assign $s_i \rightarrow x_i, \forall i;\ s_i \in S$. By implementing the partition structure and assigning a unique symbol to each time series date point, the end result was called the symbol stream. This is the re-quantized time series data that is now transformed into the symbolic domain.

2.4. **Statistical Model Development**

Once the partitions have been developed and symbols assigned to each partition, the next step is to construct the statistical model based on the resultant symbol stream. This step consists of another parameter for the SA methodology, the depth parameter *D*. The depth parameter controls the definition of model states. States in the model are formed from *D*-length subsets of symbols. Therefore, the total number of states in the algorithm given the number of partitions *P* and the depth *D* is shown in Eq. (2).

$$N_s = P^D \qquad (2)$$

As an example, assume a ternary partition scheme is implemented that results in three symbols; labeling them -1, 0, and 1. The methodology's resultant statistical states depend on the number of symbols in the algorithm as well as the chosen depth. The parameter depth adjusts the memory of the resultant symbolic model, that is, the parameter controls the groupings of symbols into states. For instance, if *D* was unity, the resultant states are 0, 1, and -1. If *D* was two, the resultant states would be 00, 01, 10, 11, 0-1, (-1)0, (-1)(-1), 1(-1), and (-1)1 according to (2).

Shown in Figure 4 is an example of the method continuing the above example with the three partition symbolic system with *D* being equal to two applied to a recorded sine wave of arbitrary amplitude. The number of resultant states is equal to three. The example sine wave in the figure is divided into zero (0), one (1) or minus one (-1) by a set threshold (partition boundary). The resultant square wave like symbol waveform developed by the processor or field programmable gate array (FPGA) is shown in the figure. The FGPA then counts the state occurrences which can then be converted into probabilities to generate what is known as the State Probability Vector (SPV).

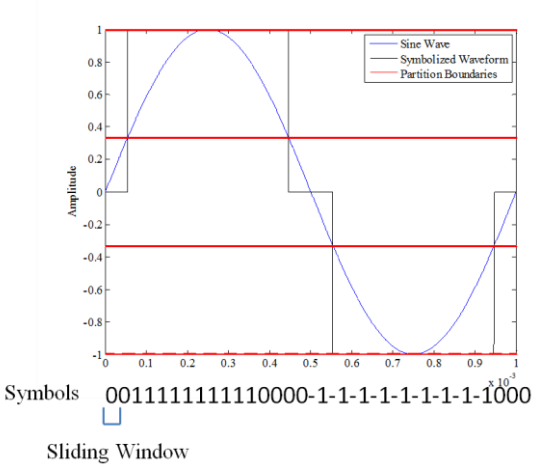


Figure 4. Example symbolization using three symbols with *d*=2 resulting in nine possible states.

With the symbol sequence s_i completed, the next step is to form states out of the symbols or groups of symbols. The probabilities of the state occurrences can be calculated and tracked across each data capture. These probabilities are arranged in a $N_s x1$ vector, where N_s represents the total number of states in the algorithm given by Eq. (2), which is the SPV. In the case where depth of the algorithm is equal to unity, as it is with most cases, the total number of states is equal to the number of symbols used. Choosing *D* equal to unity results in the smallest possible model for a given number of symbols thereby reducing computational complexity of the approach.

In addition to tracking the probability of the model states, the transition probabilities can also be calculated. The transition matrix captures the dynamics of the symbolic model and it is possible to calculate the SPV given the state transition matrix as shown in Eq. (3).

$$v_i \Pi = \lambda_i v_i \qquad (3)$$

In Eq. (3), Π is the state transition matrix, λ_i is the i^{th} eigenvalue equal to unity, and v_i is the left eigenvector of Π associated with the unity eigenvalue. Using the examples in Figure 2 and Figure 3, the state transition matrices are shown in **Table 1**.

Table 1. Example state transition matrices for uniform and ME partitioning.

Uniform Partitioning – Π Matrix		
0.99847	0.00153	0.00000
0.00278	0.99445	0.00278
0.00000	0.00153	0.99847
ME Partitioning – Π Matrix		
0.99820	0.00180	0.00000
0.00180	0.99640	0.00180
0.00000	0.00180	0.99820

Both of the matrices show little change between either types of partitioning. The results display strong diagonal terms as would be expected with symbolic analysis and with sinusoidal data. From the natural progression of the sinusoidal data, it is evident that there would be no instantaneous transitions between the minimum and maximum values resulting in the two zero transitional probabilities. The SPVs for each type of partitioning is shown in **Table 2**.

Table 2. Example SPV for Uniform and ME Partitioning.

Uniform Partitioning
0.392
0.216
0.392
ME Partitioning
0.333
0.333
0.333

The difference between uniform and ME initial SPVs can be observed in the above table. As was mentioned earlier, uniform partitioning results in equal partition sizes but not equal state probabilities. The opposite is true with ME partitioning with the resultant state probabilities equal but the partition sizes are not.

Once the probabilities or counts as shown in Table 2 are known, a distance type metric can be applied to the baseline case and future cases to develop an anomaly based on the current system operation. More deviation from this baseline will translate into a measureable anomaly at the algorithm's output.

2.5. **Anomaly Generation**

Anomalies inherent to degradation in the system can be generated from the use of the SPV between the data captures. The metric quantifies the deviation between the

known baseline, commonly known as the healthy state of the system, and a future system state. A measure commonly used to quantify an anomaly between captures is based on the Manhattan distance given in Eq. (4).

$$A = \left\| z_{nominal} - z_j \right\|_1 \quad (4)$$

In Eq. (4), $z_{nominal}$ is the nominal (baseline) SPV and z_j is the SPV at iteration j. From this measure, it is possible to quantify anomalies present in the system and how they evolve over time and usage. For the state transition matrix anomaly measure, the Frobenious norm of the difference between two state transition matrices can be used. From this evolution of the anomaly, it is then possible to define a threshold of failure for the system. The threshold can then be implemented in a predictor to estimate remaining useful life of the system.

The anomaly can be used as a diagnostic measure to determine the amount of degradation the system has incurred over its lifetime or to be used as a prognostic measure. If training data exists for the system, the anomaly measure can then be used in a prognostic application to predict the remaining useful life of the system.

3. SYNTHETIC APERTURE RADAR

The focus of the effort was in applying the Symbolic Analysis health management approach to SAR platforms. These platforms are imaging based radars that operate in frequency ranges up to the 10s of GHz. While the methodology is applicable to many systems aboard remotely piloted aircraft, the SAR platform was targeted for this research because of its importance to missions as well as the high cost of maintenance and repairs. A health methodology such as the one based on SA can reduce these costs dramatically.

The imaging radar works by mathematically assuming that a series of radar pulses and returns were generated and measured by a single large radar antenna (synthetic aperture) (Richards, Scheer, & Holm, 2010). In order to operate, the platform must be travel some finite distance during the pulse intervals.

The radar class investigated was the Active Electronically Scanned Array (AESA) radar (Melvin & Scheer, 2013). The radar itself is made up of hundreds of smaller transmit/receive (T/R) modules. Each one of these modules contains the necessary electronics for transmitting and receiving radar pulses. The T/R modules also contain the phase control block which in combination with all the other modules allows the array to electronically scan.

An example block diagram of a T/R module is shown in Figure 5. The T/R Module contains dual channels for both receiving reflections as well as for transmitting. Common to the two paths is the phase shifter for each individual element to steer the beam. The attenuator is used to add an amplitude taper to the overall array to improve the transmit characteristics. Two switches are used to select transmit and receive channels as necessary. The transmit path consists of the driver and power amp to gain the signal to the antenna element. The power is sent to the antenna through SW2 which is typically a circulator. Switching the channel to receive, the first element is the Low Noise Amplifier (LNA) with a pre-amplifier filter. The diode on the input is used to protect the LNA and for impedance matching.

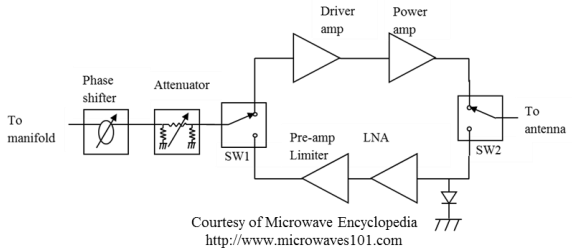


Figure 5. Example block diagram of a T/R module for an AESA radar element.

A general imaging SAR diagram is shown in Figure 6. The cross-range resolution of SAR imagery is dependent on the number of pulses sent out by the platform used in the image formation. The cross-range of a SAR image is the direction in line with the flight path of the radar system. The range direction is that which is perpendicular to the flight path. To increase range resolution, a wide bandwidth pulse is needed which would in turn require a short pulse emitted from the radar system as this short pulse would have wide bandwidth. However, to get enough signal power out such that echoes are detectable, a large instantaneous power is required which is currently unattainable with current solid-state transmitters. Instead, a frequency chirp is used so that lower instantaneous power can be used. In order to further improve the range resolution of the chirp, the resultant frequency chirp is pulse compressed.

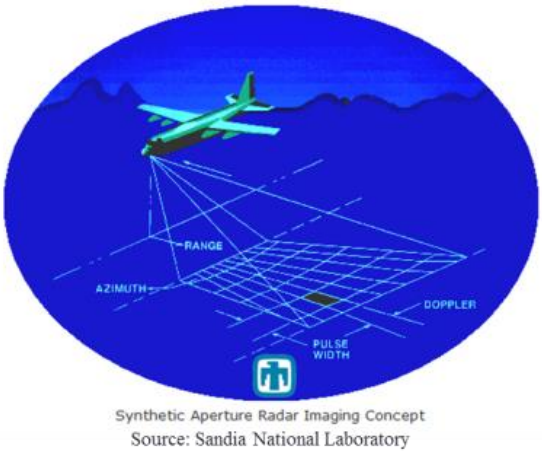


Figure 6. SAR radar imaging concept diagram.

Two types of degradation to radar images were simulated for the analysis. These events were jamming, classified as an external degradation event, and array degradation which is an internal degradation event. Both were simulated for the symbolic analysis routine. The results of these simulations were then used as input for the SA algorithm. The degradation simulations were developed to model electronic counter measures as well as deterioration effects.

The data readily available from AFRL's Sensor Data Management System (SDMS) was in the form of phase history. The phase history data is complex with both I and Q, containing both magnitude and phase of the echoes received at the radar. The phase history is calculated from the raw echo samples by using known platform related constants (flight path, etc.) and scaling (range scaling). The result is a phase history data matrix containing all N_P pulses sent from the transmitter with N_S samples per pulse. The symbolic algorithm operates on each column of the phase history matrix resulting in N_P iterations of the algorithm. The algorithm parameters must be chosen appropriately considering the number of samples available for processing and for probability convergence.

The phase history data implemented in this work was from the 2D/3D Imaging Gotcha Data Challenge ('Gotcha' dataset). This data contains phase history over 360° of azimuth of an urban environment consisting of numerous vehicles, roads, and other targets. Each degree of azimuth incorporates approximately 117 pulses with 424 frequency samples per pulse. The data was collected in the X-band (7 – 11 GHz) with a 640 MHz bandwidth. The data contains H/H, H/V, V/H, and V/V (transmit/receive) polarizations where H is horizontal and V is vertical. The different polarizations enable additional details about targets to be extracted from the reflected signals. An example from the Gotcha dataset is shown in Figure 7. The image was formed from 5° of azimuth resulting in a cross-range resolution of 0.19 m and a range resolution of .24m. The scene size is approximately 102 m by 108 m.

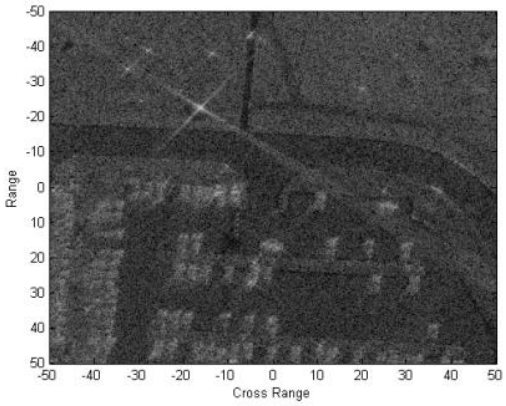


Figure 7. Example Gotcha SAR image.

The image of the parking lot located in the scene is shown in Figure 8 with the ground truth for the image in Figure 7 is shown in Figure 9**.** Figure 8 shows a view of the parking lot contained within the Gotcha scenes while Figure 9 shows the ground truth for the entire scene. The image in Figure 7 used the back projection algorithm for image generation (Gorham, & Moore, 2010). Additional photographs for the environment and targets can be found with the Gotcha Data Set (GOTCHA, 2011).

Figure 8. Parking lot image for Gotcha radar data.

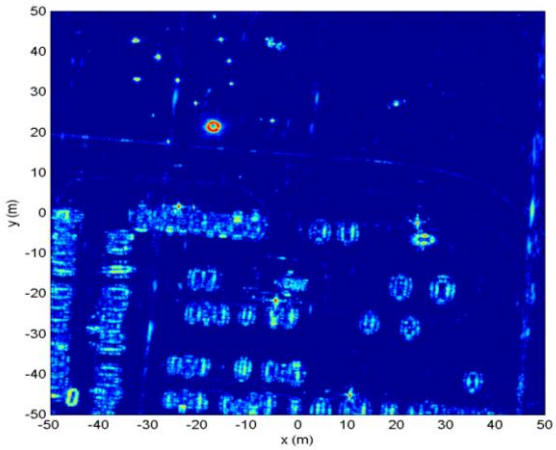


Figure 9. Gotcha ground truth.

4. RESULTS

In the Phase I work, the algorithm was simulated in a MATLAB environment investigating the SA response to both jamming and array degradation mechanisms. This section describes the approaches used to simulate the two degradation mechanisms as well as the results from the algorithm. The objective of each simulation was to determine the output of the SA algorithm to the degradation mechanisms presented in the data. In this manner, the output of the SA algorithm could also be used to intelligently classify the type of degradation (or mixture thereof) present within the system.

4.1. Jamming Degradation

The first type of degradation simulated was for radar jamming attacks. Jamming attacks are electronic countermeasures deployed to confuse or disrupt the normal operation of radar systems. There are two main types of jamming, one is related to denial of operation and the other is false target injection.

False target jamming uses an intelligent transceiver in which the source radar is monitored, manipulated, and re-transmitted. The re-transmitted signals can be used to obscure the location of ground-based objects or introduce false targets in the radar system. This type of attack falls under what is known as Digital Radio Frequency Memory (DRFM) (Kwak, 2009)(Mehalic, & Sayson, 1992)(Berger, 2001). This type of attack learns the behavior of the source radar and transmits a manipulated signal back to the receiver. The other type of attack implementing DRFM is the denial of operation. A ground based or other receiver learns the transmitted characteristics of the source radar and transmits noise at those frequencies. The transmitted noise then significantly reduces the ability to resolve objects in the image produced through SAR mapping.

Mathematically, Gaussian noise is given in Eq. (5) shown below.

$$N(\mu,\sigma) = \frac{1}{\sqrt[2]{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

In order to simulate a jamming attack and inject the additive Gaussian noise into the system, the parameters μ and σ^2 (mean and variance) must be known. These parameters are estimated from the radar data and considered as the healthy non-degraded parameters. With the parameters defined, the noise is added into the system as shown in Eq. (6).

$$PH_{corrupted} = PH_{original} + N(\alpha\mu_0, \alpha\sigma_0) + jN(\alpha\mu_0, \alpha\sigma_0) \quad (6)$$

In (6), PH is the Phase History, α is a scalar, and $N(\mu,\sigma)$ is the additive Gaussian noise. Note that in Eq. (6), the noise is added to both the real and imaginary components of the PH. Each additive noise component is independent of each other. The scalar, α, is defined in Eq. (7).

$$\alpha = 10^{\frac{P(dB)}{20}} \quad (7)$$

The parameter controls the strength of the jamming attack such that if P(dB) = 0, the Signal-to-Noise Ratio (SNR) of the resultant system would be 0 dB. The resultant power of the jamming noise would equal to that of the returned echoes.

The jamming corruption was then implemented on the Gotcha data set. In this case, $P(dB)$ was chosen to be 0 dB. The estimated noise parameters are shown in Table 3.

Table 3: Estimated Noise Parameters from Gotcha Radar Data

Estimated Noise Parameters from Radar Data		
	Real	**Imaginary**
Mean, μ	2.450e-7	8.373e-8
Variance, σ_v^2	6.461e-4	6.461e-4

This results in the scalar, α, having the value of unity. The resulting image is shown in Figure 10**.**

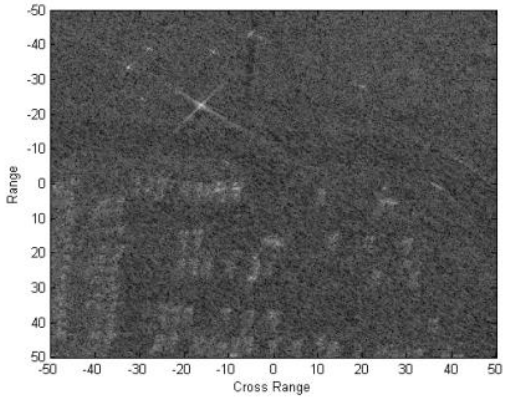


Figure 10. Jamming corruption: Gotcha SAR image.

Compare the results of Figure 10 to those in Figure 7 which contain the original image. As anticipated, the jamming significantly reduces the ability to resolve objects in the image. The stronger reflections in the scene due to metallic objects can still be seen due to the starburst effect; however, the details of the road and parking lot are significantly reduced.

The PH data with the included jamming noise was then implemented in the SA algorithm. The parameters used in the analysis are shown in **Table 4.**

Table 4: SDAAD Parameters for ME and Uniform Partitioning – Jamming

Parameters	**Number of Partitions**	**Depth**	**Resultant Number of States**
Uniform Partitioning	6	1	6
Maximum Entropy	6	1	6

For all of the following results, the SA routine was implemented on the magnitude of the PH data. The magnitude was chosen as it would represent any change between both the real part and the imaginary component of the PH. Other features that could be used are the individual

real or imaginary components or the angle between the real and imaginary components.

4.1.1. **Jamming – Uniform Partitioning**

The first set of results was developed with uniform partitioning. The resultant anomaly for the uniform partitioning jamming attack is shown in Figure 11. Recall that the signal to noise ratio (SNR) of this system was simulated to be 0dB in order to simulate a significant strength jamming attack to the radar platform.

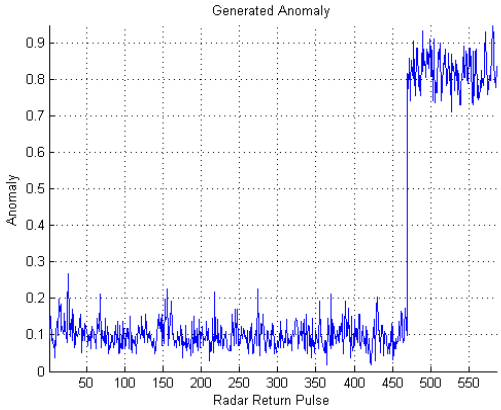


Figure 11. Jamming corruption: anomaly results – uniform partitioning.

In the figure, the jamming attack is clearly seen in the last 117 pulses of the image. In addition, the effects of jamming on these pulses, which represent about 20% of the total image, were shown in Figure 10. If the entire group of return pulses had been jammed, the image would have been totally corrupted but in order to demonstrate the change from a jammed pulse to a non-jammed pulse only the last 117 return pulses were jammed. The resulting anomaly has a magnitude of about 0.85. A threshold could be implemented around an anomaly magnitude of 0.8 to detect this type of degradation.

4.1.2. **Jamming – ME Partitioning**

The resultant anomaly magnitude formed from the state probabilities using the anomaly measure is shown in Figure 12.

Comparing these results to those obtained from the uniform partitioning, they are both similar in that both partitioning methods detect the added jamming noise at the instance it was injected. The resultant magnitude of the anomalies is also comparable at about 0.85. A notable difference is in the anomaly measure before the jamming. As can be observed in the ME partitioning, the anomaly is slightly larger.

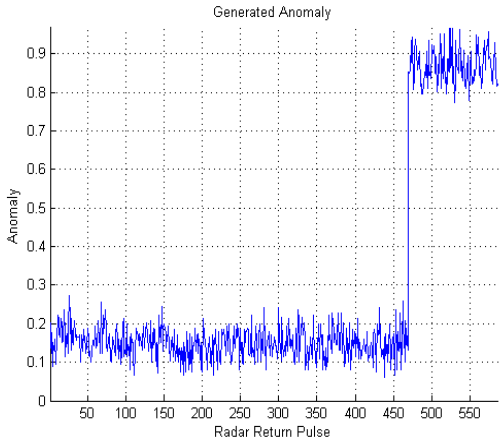


Figure 12. Jamming corruption: anomaly results – ME partitioning.

Recall that ME partitioning results in partition structures that finely divide dense regions of data and coarsely divide sparse regions. This also results in equal initial partition probabilities and hence symbol probabilities that evolve with degradation. Due to this distribution, any small deviation, either from degradation or environment, can be detected by this partitioning methodology. Figure 12 shows a slightly larger anomaly magnitude which is a result of slight differences in data between pulses. This slight increase may be problematic when the approach is applied to a data from a fielded system. Because of this, uniform partitioning may be the most appropriate partition approach for future work.

4.2. **Array Degradation**

Array degradation was the next type of degeneration that was simulated. This type of degradation represents internal platform degradation and was also implemented using the SAR Gotcha dataset. From the T/R module (Figure 5), there are two paths within each array module. The weakest link in each module is the power amplifier used as the final stage to drive the antenna. It was assumed in this analysis the amplifier fails such that the module can no longer transmit. Since the receive path is still intact, it is assumed that the module can receive echoes.

If the amplifier fails and the receive path is still active the overall transmit power decreases but the receive gain remained the same. It is known that the output power of an array degrades according to Eq. (8) (Rutledge, Cheng, York & Weikle, 1999).

$$dB_{LOSS} = 20log_{10}(1-\beta) \qquad (8)$$

The total transmit power loss can then be related to the percentage of failed elements β. The received power derived from the radar range equation is given in Eq. (9) (Richards et al, 2010).

$$P_r = \frac{P_t G_t G_r \lambda^2 \varepsilon}{4\pi^3 R^4} \quad (9)$$

In (9), P_t is the power transmitted, G_t is the gain of the transmit antenna, G_r is the gain of receive antenna, λ is the carrier wavelength, ε is related to the target's radar cross section (RCS), and R is the range to the target. In normal radar operation, the same antenna receives and transmits resulting in the same gain. However, the loss in transmit power can be modeled by applying a scalar directly to G_t which then directly results in a decrease in the received power since it is assumed that the receiver gain remains constant due to the fact that all elements can functionally receive echoes.

The transmit gain during the degradation simulation is shown in Figure 13**.** As was done with the jamming simulation, the array degradation was applied to 117 individual pulses on a single degree of azimuth. In this manner, each pulse was scaled by the values of the linear relationship shown in Figure 13.

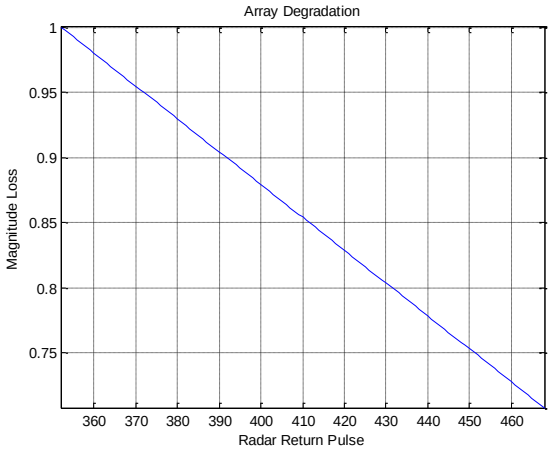


Figure 13. Array degradation simulation: Transmit gain plot, G_t for use in Eq. (9).

The scaling in the figure results in an applied -3 dB transmit loss to the antenna. This level was chosen as it is considered the failure point for a transmitting antenna. A 3dB loss translates to approximately 29% of the element modules failing in the array. An image formed from a simulated degraded array is shown in Figure 14.

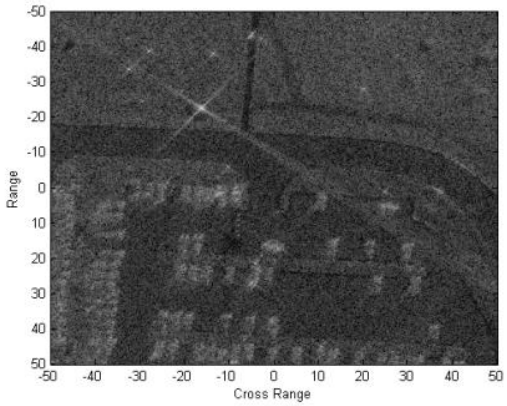


Figure 14. Array degradation: Gotcha SAR image.

The image degradation is minimal compared to the original non-degraded image shown in Figure 7. The image details of the parking lot can still be seen in the degraded image including the roadways and parked vehicles. For the SA analysis, the parameters implemented are shown in Table 5. The parameters implemented were the same as was implemented in the jamming simulation.

Table 5: SDAAD Parameters for ME and Uniform Partitioning – Array Degradation

Parameters	**Number of Partitions**	**Depth**	**Resultant Number of States**
Uniform Partitioning	6	1	6
Maximum Entropy	6	1	6

4.2.1. **Array Degradation – Uniform Partitioning**

The resultant anomaly formed from the deviation of these states from the baseline is shown in Figure 15. The figure shows the increasing anomaly that follows the degradation profile simulated. Note that the return pulse numbers in Figure 13 coincide with the algorithm output return pulse numbers in Figure 15.

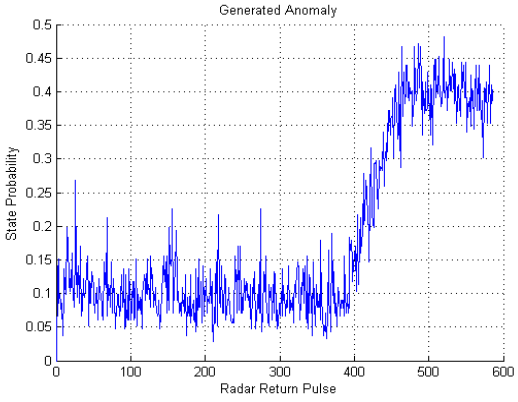


Figure 15. Array degradation: anomaly results– uniform partitioning.

For this simulation, the degradation profile was simulated on the second to last azimuth angle again applied to 117 pulses. The last azimuth angle was maintained at the -3dB degradation level. The increase in anomaly is observable and when the degradation is constant, the resultant anomaly is constant as well. The result also demonstrates the possibility of implementing a remaining useful life predictor on this type of degradation. This would assume that the array would degrade slowly over its useful life before needing to be pulled from the platform for repair. Through these simulations, the anomaly magnitude from a jamming event resulted in a larger anomaly magnitude which was due to the simulation. For example, weaker jamming attempts or more array degradation could result in comparable anomaly magnitudes. In future work these situations will be resolved by the classifier stage. In addition, the past history of the algorithm output can be used to discriminate between wear-out phenomenon in the array and deliberate platform jamming.

4.2.2. **Array Degradation – ME Partitioning**

The resultant anomaly formed from the same simulation using ME partitioning is shown in Figure 16**.**

As was done with the simulation under uniform partitioning, the degradation was applied to the second from the last azimuth degree so that the final degree could be held at the -3dB array degradation level. The resultant anomaly plot was similar to that obtained with uniform partitioning and the resultant magnitudes are also comparable. In this case, the -3dB anomaly magnitude is only slightly larger due to the larger nominal anomaly from pulses 1 through 400 (~0.15 for ME to ~0.10 for Uniform).

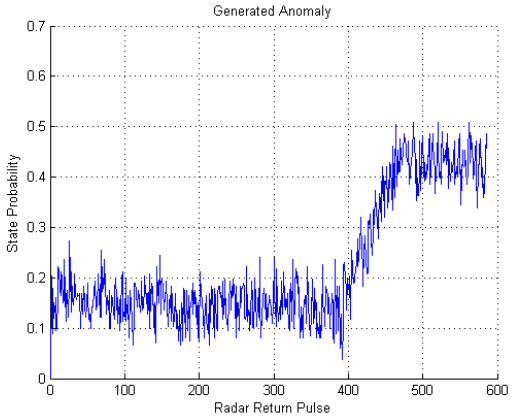


Figure 16. Array degradation: anomaly results – ME partitioning.

This result was also observed with the jamming results of the previous section. The difference is slight and the resultant responses from the partitioning methods remain similar. Since the results are similar, the application of this approach to SAR platforms would dictate that either partitioning method could be implemented. Obtaining more data from fielded system may give more insight into which approach would be more applicable for degradation monitoring. From this initial research, although the results are positive in general, a determination of which partitioning methodology is superior to the other cannot be stated.

5. **COMBINED DEGRADATION**

Separate degradation mechanisms such as those above can be easily identified when they occur by themselves. More interesting is the case when multiple degradation mechanisms occur simultaneously. For this reason, the two degradation mechanisms above were simulated simultaneously with the effects superimposed in the data. For instance, the array was first degraded by applying the degradation to the second from the last azimuth angle of data and holding the last angle of data at -3dB degradation. At this point, a jamming attack was simulated on top of the array degradation.

In this case, the parameters for the test were, six partitions, depth of unity, and the partitioning method was uniform. In this case, uniform was arbitrarily chosen since each approach yielded similar results in the previous analysis. The resultant SAR image formed from the combined degradation is shown in Figure 17. As with previous simulations, the jamming power was again set to 0dB.

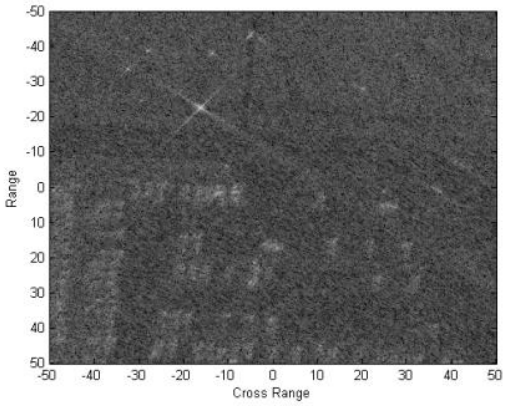


Figure 17. Combined degradation: Gotcha SAR.

In the image, the degradation is observable with the jamming being the strongest source of degradation. Compare this image to that obtained from jamming only, Figure 10. The two images look similar with Figure 17 showing slightly more image degradation. The resultant anomaly from this combined effect is shown in Figure 18.

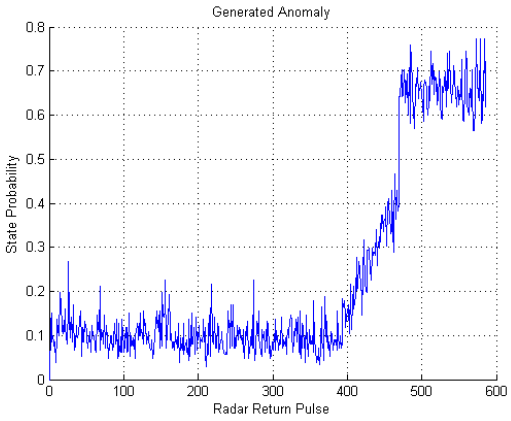


Figure 18. Combined degradation: anomaly results– uniform partitioning.

The results in Figure 18 show a distinct combination of the two degradation effects. In pulses 400 through 480, the array degradation is clearly seen. In pulses 480 through 580, the combined effects of jamming and array degradation are seen although the strength of the jamming attack overcomes that of array degradation and manifests itself as a discontinuity in the anomaly magnitude. The discontinuity that arises from jamming attacks could be implemented in the degradation classifier and assist in determining whether degradation is internal or external.

6. CONCLUSION AND FUTURE WORK

The method of Symbolic Analysis was demonstrated using simulated degradation in SAR phase history data. Under a MATLAB environment, both jamming and array degradation were simulated and the results observed. The simulations analyzed the results from both uniform and maximum entropy partitioning methods under the same number of partitions and algorithm depth. The data used was phase history data that was corrupted with degradation representing jamming and array failure events. Once corrupted, the magnitude of this data was used as the input into the SA algorithm. The results show similarities between the two with ME being slightly more sensitive to the data as compared to uniform. In addition, the results were simulated with combined degradation mechanisms. In these cases, it was shown that it is possible to perform classification on the resultant algorithm output such that degradation can be identified. From these initial results, it seems to be the case that uniform partitioning would be preferable to ME to reduce the probability of false positives.

QorTek has been awarded a Phase II research program to expand the methodology and apply to both healthy and degraded field data from imaging radars. The new research project will investigate the results of the Phase I to validate the simulations as well as to expand the number of degradation mechanisms to model. Another objective of this research is to expand on the degradation classification as well as investigate the application of prognostics to the approach. QorTek plans to also use this research to definitively determine if there is a superior partitioning methodology between the two presented in the initial work. In addition, the Phase I work only investigated using the magnitude and not the angle of the complex data. The Phase II work will investigate using additional features and using the partitioning approach to generate a one-dimensional symbolic data set. It is anticipated that the algorithm will be implemented and a prototype flight-tested on a SAR radar payload.

The output of the SA can be utilized in a prognostic application. The output of the algorithm would provide a measurement of degradation which would act as an input for a Kalman-type predictor. As was observed, the output of the algorithm is related to the amount of degradation sustained by the radar. Since the exact evolution of the radar faults are not exactly known, a generic model must be implemented for the Kalman filter. A kinematic-motion model could be applied for the Kalman model. Future work will also address the determination of how much degradation can be sustained by the payload until it is deemed 'failed.' This work is anticipated to be carried out in the Phase II program.

ACKNOWLEDGMENTS

This work was funded by the US Air Force under contract FA8650-11-M-3138. QorTek would like to express our thanks to Thierry Pamphile from RQQI and Dale Kristof from WIIE for their support during this SBIR work. QorTek would also like to acknowledge the assistance

from LeRoy Gorham, RYAPX, Wright-Patterson Air Force Base. His help with SAR image processing is greatly appreciated.

NOMENCLATURE

A	=	anomaly
α	=	noise scaling constant
β	=	percentage of failed array
D	=	symbolic depth
$H(\cdot)$	=	entropy
M	=	time series data length
N_s	=	number of states
$p(\cdot)$	=	probability
P_i	=	i^{th} partition
s_i	=	i^{th} symbol
U	=	time series data amplitude range
X	=	time series data
z	=	state probability vector
v_i	=	i^{th} eigenvector
λ_i	=	i^{th} eigenvalue
Π	=	state transition matrix

REFERENCES

Bower, G., Mayer, J., & Reichard, K. (2011). "Symbolic Dynamics and Analysis of Time Series Data for Diagnostics of a dc-dc Forward Converter," in *Annual Conference of the Prognostics and Health Management Society*, Montreal, 2011.

Bower, G., Mayer, J., & Reichard, K. (2008). "Symbolic Dynamics for Anomaly Detection in a dc-dc Forward Converter," *Proceedings of the 2008 International Conference on Prognostics and Health Management*, Denver, October 2008.

Berger, S. D. (2001). "The spectrum of a digital radio frequency memory linear range gate stealer electronic attack signal," *Proceedings of the 2001 IEEE Radar Conference*, Atlanta, 2001.

Daw, C.S., C.E.A. Finney & E.R. Tracy (2003). "A review of symbolic analysis of experimental data." Review of Scientific Instruments 74.2 (2003): 915-930.

Gorham, L. & L. Moore (2010). "SAR image formation toolbox for MATLAB." *Proceedings of the SPIE*. 2010.

Kwak, C.M. (2009). "Application of DRFM in ECM for pulse type radar," *34th International Conference on Infrared, Millimeter, and Terahertz Waves*, Busan, Korea, 2009.

Mehalic, M. & Sayson, A. M. (1992). "A dual-port DRAM component for a digital RF memory," *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference*, Dayton, 1992.

Melvin, William L., Scheer, James, A. (2013). *Principles of Modern Radar: Advanced Techniques*. Edison, NJ: SciTech Publishing.

Papoulis, A., & Pillai U. S. (2002). *Probability, Random Variables, and Stochastic Processes.* 4th Edition, New York, NY: McGraw-Hill.

Ray, Asok (2004). "Symbolic dynamic analysis of complex systems for anomaly detection." *Signal Processing* (2004): 1115-1130.

Richard, Mark A., Scheer, James A., & Holm, William A. (2010). *Principles of Modern Radar: Basic Principles*. Raleigh, NC: SciTech Publishing, Inc.

Rutledge, D., Cheng, N., York, R. & Weikle II, R. (1999). "Failure of Power Combining Arrays." *IEEE Transactions on Microwave Theory and Techniques* 47.7 (1999): 1077-1082.

U.S. Air Force. Gotcha Volumetric SAR Data Set. Sensor Data Management System. August 2011. <https://www.sdms.afrl.af.mil/index.php?collection=gotcha>.

BIOGRAPHIES

Gregory Bower received his B.S., M.S, and Ph.D. degrees all from the Pennsylvania State University. He is currently the Chief Technology Officer for QorTek, Inc. where he focuses on diagnostic and prognostic methods and power electronic design and development. Previously, he was a research assistant for the Applied Research Laboratory in State College, PA. His research interests include Prognostics and Health Management (PHM) of electronic systems, statistical methods, robust and optimal control theory, system identification, and power conversion. He is a member of Eta Kappa Nu, Tau Beta Pi, and IEEE.

Jonathan Zook received a B.S. in Electronics Engineering from Pennsylvania State University. Jonathan is the Engineering Manager of QorTek, Inc. responsible for helping to oversee innovative projects on the cutting edge of power electronics design. His research interests include high speed parallel processing, digital control of high efficiency power converters and amplifiers for smart material actuators. He is a member of the IEEE.

Ross Bird received his B.S. in Electronics from Penn State in 2001 and received his MSEE in 2003. He is currently the president of QorTek, Inc. since 2010. He has worked extensively at the leading edge of power electronics design holding several patents that incorporate advanced materials, design and digital control within these systems. His research interests include wide bandgap power electronics and devices, high efficiency power conversion, extremely high density power electronics, smart materials systems, and piezoelectronics. He is a member of the IEEE.

Learning Diagnosis Based on Evolving Fuzzy Finite State Automaton

Moussa Traore[1], Eric Châtelet[2], Eddie Soulier[3], and Hossam A. Gabbar[4]

[1,2,3] *University of Technology of Troyes (UTT), Troyes, BP 2060, 10010, France*
moussa_amadou.traore@utt.fr
eric.chatelet@utt.fr
eddie.soulier@utt.fr

[4] *Faculty of Energy Systems and Nuclear Science, University of Ontario Institute of Technology, 2000 Simcoe St. North, Oshawa, Ontario, Canada L1H7K4*
Hossam.Gaber@uoit.ca

ABSTRACT

Nowadays, determining faults (or critical situations) in non-stationary environment is a challenging task in complex systems such as Nuclear center, or multi-collaboration such as crisis management. A discrete event system or a fuzzy discrete event system approach with a fuzzy role-base may resolve the ambiguity in a fault diagnosis problem especially in the case of multiple faults (or multiple critical situations). The main advantage of fuzzy finite state automaton is that their fuzziness allows them to handle imprecise and uncertain data, which is inherent to real-world phenomena, in the form of fuzzy states and transitions. Thus, most of approaches proposed for fault diagnosis of discrete event systems require a complete and accurate model of the system to be diagnosed. However, in non-stationary environment it is hard or impossible to obtain the complete model of the system. The focus of this work is to propose an evolving fuzzy discrete event system whose an activate degree is associated to each active state and to develop a fuzzy learning diagnosis for incomplete model. Our approach use the fuzzy set of output events of the model as input events of the diagnoser and the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment.

1. INTRODUCTION

A great number of systems or situations can be naturally viewed as discrete event systems. A discrete event system is a dynamic system whose the behavior is governed by occurrence of physical events that cause abrupt changes in the state of the system (Liu & Qiu, 2009a; Cassandras & Lafortune, 1999; Moamar & Billaudel, 2012; Traore, Moamar, & Billaudel, 2013). Discrete event system theory, particularly on modeling and diagnosis, has been successful employed in many areas such as concurrent monitoring and control of complex system (Cao & Ying, 2005). Usually, a discrete event system is modeled by Automaton (Dzelme-Berzina, 2009; Mukherjee & Ray, 2014) or Petri Net (Patela & Joshi, 2013). Automaton (or more precisely a finite state automaton) are the prime example of general computational systems over discrete spaces and have a long history both in theory and application (Thomas, 1990; Moghari, Zahedi, & Ameri, 2011). A finite state automaton is an appropriate tool for modeling systems and applications which can be realized as finite set of states and transition between them depending on some input strings (Doostfatemeh & Kremer, 2004). And, the behavior of discrete event system modeled by an automaton is described by the language generated by the automaton.

Discrete event systems are divided into two categories: crisp discrete event system and fuzzy discrete event system. A crisp discrete event system is usually described by a deterministic automaton (Luo, Li, Sun, & Liu, 2012) and fuzzy state is the extension of crisp discrete event system by proposing fuzzy state and every state transition is associated with a possibility degree, called in the following membership value. Thus, the membership value can be defined as the possibility of the transition from current (active) state to next state. The main advantage of fuzzy finite state automaton is that their fuzziness allows them to handle imprecise and uncertain data, which is inherent to real-world phenomena, in the form of fuzzy states and transitions. In literature, many application of fuzzy discrete event system had been proposed (Gerasimos, 2009; Luo et al., 2012; Sardouk, Mansouri, Merghem-Boulahia, & Gaiti, 2013). Thus, one of the interesting characteristics of fuzzy automaton is the possibility of several transitions from different current fuzzy states lead to the same next fuzzy state simultaneously, and also the possibility of several transitions

Moussa Traoré et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

from one current fuzzy state lead to the different next fuzzy states simultaneously and consequently several output label can be activated at the same time (Doostfatemeh & Kremer, 2005). For this reason, fuzzy discrete event is very adapted to resolve the ambiguity in a fault diagnosis problem especially in the case of multiple faults. In this paper, these output events constituted of a fuzzy set are applied as input event for our diagnoser. Most of applications, the output should be crisp. Therefore, the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment. Thus, the outputs are assumed to be observable.

The diagnosis of discrete event systems is a research area that has received a lot of attention in the last years and has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems (Cabasino & Alessandro Giua, 2010) or complex situation (like crisis situation) (Traore et al., 2013). Hence, the use of finite state automaton in fault diagnosis tasks has gained particular attention in the case of discrete event dynamic systems (Gerasimos, 2009). Although, most of approaches proposed in literature for fault diagnosis of discrete event systems require a complete and accurate model of the system to be diagnosed. However, the discrete event model may have arisen from abstraction and simplification of a continuous time system or through model building from input-output data. As such, it may not capture the dynamic behavior of the system completely. Therefore, in this paper, we attempt to develop a diagnosis approach based on fuzzy automaton for incomplete model in non-stationary environment. For most of real-world applications operate in non-stationary environment.

The diagnosis approach proposed in our paper is different from the approach proposed in (Kwong & Yonge-Mallo, 2011). In our paper, the diagnoser is a finite-state Automaton which takes fuzzy output sequence of the system as its input. Here, the learning diagnoser is constructed off-line and the diagnosis is performed on-line using input and output data generated by system's model. The on-line diagnosis system allows to build an evolving fuzzy finite state system by updating the set of states and/or the set of input symbols. The new states and/or transitions detected by the diagnoser is validated by an expert of the system or situation.

The potential application of learning diagnosis based on fuzzy finite state automaton is in solving the ambiguity in a fault diagnosis problem especially in the case of multiple faults.

This paper is organize as follows. In section 2 , we present the required background of crisp discrete event system. We describe the general definition for fuzzy discrete event system in section 3 . The standard diagnoser is presented in section 4. The algorithm of the learning diagnosis based on evolving fuzzy finite state automaton is proposed in section 5. Learning diagnoser application to crisis management is presented in section 6.

2. CRISP DISCRETE EVENT SYSTEM

A crisp discrete event system is usually described by a deterministic automaton $G = \{X, \Sigma, \varphi, Y, x_0, F\}$, where

- X is the set of states
  $$X = \{x_0, x_1, \cdots, x_{n-1}, x_n\},$$
- Σ is set of input symbols,
  $$\Sigma = \{a_0, a_1, \cdots, a_{m-1}, a_m\},$$
- $\varphi : X \times \Sigma \to X$ is the transition function,
- Y is the set non-empty finite set of output,
  $$Y = \{y_0, y_1, \cdots, y_{l-1}, y_l\},$$
- $x_0 \in X$ is the start state and
- $F \subseteq X$ is the (possibly empty) set of accepting or terminal states,

The event set Σ includes the set of failure events (or critical events) Σ_f (Kwong & Yonge-Mallo, 2011). In addition to the normal situation (mode) N, there are p critical situation (or failure mode) $F_1, \cdots, F_p$ that describe the evolution of the condition's system. We denote the condition set of the situation by
$\lambda = \{N, F_1, \cdots, F_p\}$, in this case, the state set partitioned into

$$X = X_N \cup X_{F_1} \cup \cdots \cup X_{F_p}.$$

In (Traore et al., 2013), we proposed the extension of the transition function φ represented as: $\varphi : X \times \Sigma \to X \times Y$.
Let φ_1 and φ_2 be the two projection of φ such as φ_1 gives the state reached from a state $x_i \in X$ and a given input $a_k \in \Sigma$ and φ_2 defines the output sequence from state x_i and input a_k. The expression of φ_1 and φ_2 are given by

$$\varphi_1(x_i, a_k) = \{x_j \mid \exists\, y_j \text{ such that } (x_j, y_j) \in \varphi(x_i, a_k)\},$$
$$\varphi_2(x_i, a_k) = \{y_j \mid \exists\, x_j \text{ such that } (x_j, y_j) \in \varphi(x_i, a_k)\},$$

where x_i, $x_j \in X$ and $a_k \in \Sigma$ and $y_j \in Y$. The new definition of φ is:

$$\varphi(x_i, a_k) = (\varphi_1(x_i, a_k), \varphi_2(x_i, a_k)).$$

These two projection may be extended to take input sequence, for example: $x_j \in \varphi_1(x_i, \sigma_i \in \Sigma^*)$ and/or output sequence for example: $\sigma_y \in \varphi_2(x_i, \sigma_i \in \Sigma^*)$, where $\sigma_i = a_1 a_2 \cdots a_l$ and $\sigma_y = y_0 y_1 \cdots y_n$. Σ^* is a set of all strings formed by events in Σ, example $a_k \in \Sigma$, then, $a_1 a_2 \cdots a_k \in \Sigma^*$.

The behavior of G is described by the language generated by G denoted as $\mathscr{L}(G)$ or simply by $\mathscr{L}$ (Liu & Qiu, 2009b).

3. FUZZY DISCRETE EVENT SYSTEM

Fuzzy discrete event systems as a generalization of (crisp) discrete event systems have been introduced in order that it is possible to effectively represent uncertainty, imprecision, and vagueness arising from the dynamic of systems. A fuzzy

discrete event system has been modelled by a fuzzy automaton; its behavior is described in terms of the fuzzy language generated by the automaton (Cao & Ying, 2006).

A Fuzzy Finite Automaton (FFA) is a 6-tuple

$$\widetilde{G} = \{X, \Sigma, \delta, Y, \widetilde{x}_0, F\}.$$

i The fuzzy subset $\delta : X \times \Sigma \times X \to [0\ 1]$ is a function, called the fuzzy transition function. A transition from state x_i (current state) to x_j (next state) upon a_k with the weight ω_{ij} is denoted as: $\delta(x_i, a_k, x_j) = \omega_{ij}$,

ii $\widetilde{x}_0 \in X$ is the set of initial states.

One of the interesting characteristics of FFA is the possibility of several transitions from different current (or active) states lead to the same next state simultaneously (*see* Figure 1.(a)). Thus, the possibility of several transitions from one current states lead to the different next states simultaneously as shown in Figure 1.(b), and consequently several output label can be activated at the same time (Doostfatemeh & Kremer, 2005). It is possible to have more than one start state with FFA.

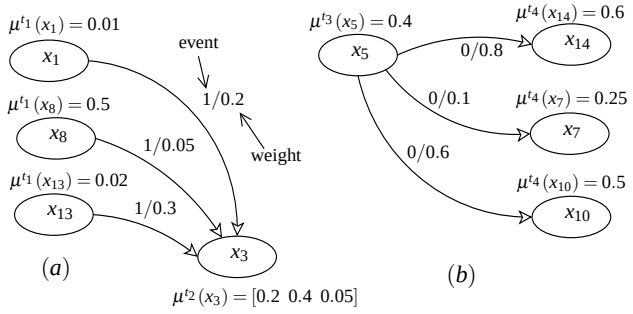


Figure 1. A example of FFA.

when an input a_k occurs at time t, all active state at this time, are those states to which there is at least one transition on the input event a_k. Then, the fuzzy set of all active state at time t is called active state set at time t. A active state set denoted X_{act} is consisted of state and their $mv's$. The definition of X_{act} is given by:

$$X_{act}(t) = \left\{ \left(x_j, \mu^t(x_j)\right) | \exists (x_i \in X_{pred}(x_j), a_k \in \Sigma) \wedge x_j \in X_{succ}(x_i, a_k) \right\},$$
$$X_{pred}(x_j) = X_{pred}(x_j, t) \text{ and,}$$
$$X_{pred}(x_j, t) = \left\{ x_i \mid \exists\ a'_k \text{ s.t } x_j \in \varphi_1(x_i, a'_k) \ \wedge \ x_j \in X_{act}(t) \right\},$$
$$X_{succ}(x_i, a_k) = \left\{ x_j \mid x_j \in \varphi_1(x_i, a_k) \right\},$$
$$\delta(x_i, a_k, x_j) = \omega_{ij},$$

For example in Figure 1.(a)
$\varphi_1(x_1, 1) = \varphi_1(x_8, 1) = \varphi_1(x_{13}, 1) = x_3$.

where x_i is the state at time $t-1$, $\mu^t(x_j)$ is the membership of state x_j at time t, $X_{pred}(x_j, t)$ is all predecessors set of active state x_j and $X_{succ}(x_j, a_k)$ is all successors set of the state x_j on input symbol a_k. The successor $X_{succ}(x_j, a_k)$ is the set of all x_j which will be reached via transition function $\delta(x_j, a_k)$. In the following, all successors set of x_j is denoted by $X_{succ}(x_j, \overset{all}{\to})$, when the next state depend to the occurrence of different events.

We use the same notation for the active state, when the upon entrance is a string Γ. The active state set of the string Γ is given by:

$$X_{act}(\Gamma) = X_{act}(t_0 + |\Gamma|),$$

where $|\Gamma|$ represent the length of Γ.

Definition 1 *A fuzzy set Δ_X defined on a set X (discrete or continuous), is a function mapping each element of X to a unique element of the interval $[0\ 1]$, $\Delta_X : X \to [0\ 1]$. The membership value (mv) of the state $x_i \in X$ at time t is denoted as $\mu^t(x_i)$.*

For example in Figure 1.(a), at time time t_1, the active state is $X_{act}(t_1) = \{x_1, x_8, x_{13}\}$ and $X_{succ}(x_1, 1) = \{x_3\}$, $X_{succ}(x_8, 1) = \{x_3\}$ and $X_{succ}(x_{13}, 1) = \{x_3\}$, and at time time t_2, the active state is $X_{act}(t_2) = \{x_3\}$ and $X_{pred}(x_3, t_2) = \{x_1, x_8, x_{13}\}$, that mean the state x_3 is forced to take several different mv at this time. Hence, x_3 is a state with multi-membership, that we will call in the following multi-membership state.

In Figure 1.(b), each mv $\mu^{t+1}(x_j)$ of the state x_j at time $t+1$ is computed by using the function Ψ_1, named augmentation transition function. The function Ψ_1 should satisfy the two following axioms.

1. $0 \leq \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right) \leq 1$,
2. $\Psi_1(0, 0) = 0$ and $\Psi_1(1, 1) = 1$.

To compute $\mu^{t+1}(x_j)$, the function Ψ_1 use two parameters: $\mu^t(x_i)$ at time t and the weight ω_{ij} of the transition.

same example of Ψ_1 are:

- Arithmetic Mean

$$\begin{aligned} -\mu^{t+1}(x_j) &= \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right), \\ &= Mean(\mu^t(x_i), \omega_{ij}), \\ &= \frac{\mu^t(x_i) + \omega_{ij}}{2}, \end{aligned}$$

- Geometric Mean

$$\begin{aligned} -\mu^{t+1}(x_j) &= \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right), \\ &= GMean(\mu^t(x_i), \omega_{ij}), \\ &= \sqrt{\mu^t(x_i) \times \omega_{ij}}, \end{aligned}$$

where $\mu^t(x_i)$ is the mv of the corresponding predecessor of x_j and $\delta(x_i, a_k, x_j) = \omega_{ij}$.

The mv of each active state is used as the level of activation of each active state and the active state can be multi-membership state. However, in this paper, we need a single value for each active state. For this reason, the function Ψ_2 is introduced

to compute the single *mv* corresponding to the state that was forced to take several *mv* by these predecessors. The single membeship value $\mu^{t+1}(x_j)$ of each multi-membership state given by:

$$-\mu^{t+1}(x_j) = \overset{m}{\underset{i=1}{\Psi_2}}[\Psi_1(\mu^t(x_i),\omega_{ij})],$$

where m is the number of simultaneous transitions from states x_i to state x_j prior to time $t+1$.

The function Ψ_2 should satisfy the minimum requirements following axioms:

1. $0 \leq \overset{m}{\underset{i=1}{\Psi_2}}[\Psi_1(\mu^t(x_i),\omega_{ij})] \leq 1,$

2. $\Psi(\phi) = 0,$

3. $\overset{m}{\underset{i=1}{\Psi_2}}[\Psi_1(\mu^t(x_i),\omega_{ij})] = \nu$, if $\forall(\Psi_1(\mu^t(x_i),\omega_{ij}) = \nu),$

same example of Ψ_2 are:

- Maximum multi-membership resolution

$$-\ \mu^{t+1}(x_j) = \underset{i=1\ to\ m}{Max}\left[\Psi_1(\mu^t(x_i),\omega_{ij})\right],$$

- Arithmetic mean multi-membership resolution

$$-\ \mu^{t+1}(x_j) = \frac{\left[\sum_{i=1}^{m}\Psi_1(\mu^t(x_i),\omega_{ij})\right]}{m},$$

4. CASE STUDY

Consider the *FFA* in Figure 4 with several transition overlaps and several output labels. It is specified as:

$$\widetilde{G} = (X,\Sigma,\delta,Y,\widetilde{x}_0,F),$$

The dashed line in Figure 4, between states 12 and 13 represents a failure event or critical event. The occurrence of event $''f''$ bring the system in failure (or critical) mode corresponding to state x_{13}.

For instance, during the crisis management, the procedures designed by one or more organizations for the crisis situations can be applied, or partially applied or no applicable (no suitable) for the current situation. This latter case can be modeled by the state x_{13} in Figure 4 and for the reconfiguration, the model of crisis must be evolving and accepting missing information, whose the advantage to develop an evolving fuzzy finite state automaton for crisis management.

In this example

$$\begin{aligned}
X &= \{x_0,x_1,\cdots,x_{13}\},\ \text{the set of states},\\
\Sigma &= \{a,b,c,d,e\},\ \text{set of input symbols},\\
Y &= \{\theta,\alpha,\beta,\gamma,\mu,\rho,\kappa,\xi,\eta\},\ \text{set of output},\\
\widetilde{x}_0 &= \left\{x_0,\mu^{t_0}(x_0)\right\},\ \text{fuzzy subset initial state},\\
\Delta_X &= \{0.04,0.09,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\},
\end{aligned}$$

$$\lambda(x_i) = \begin{cases} F_1, & \text{if i=13},\\ N, & otherwise \end{cases}$$

we suppose, $\mu^{t_0}(x_0) = 1$ at the beginning and $\widetilde{x}_0 = \{(x_0,1)\}$ and all the other *mv* are computed by using the function Ψ_2 and/or Ψ_1.

Assuming that $\widetilde{G}$ starts operating at time t_0 and the next three input are **"a, e, d"** respectively (one at a time), active states and their *mv*$'$*s* at each time step are as follows.

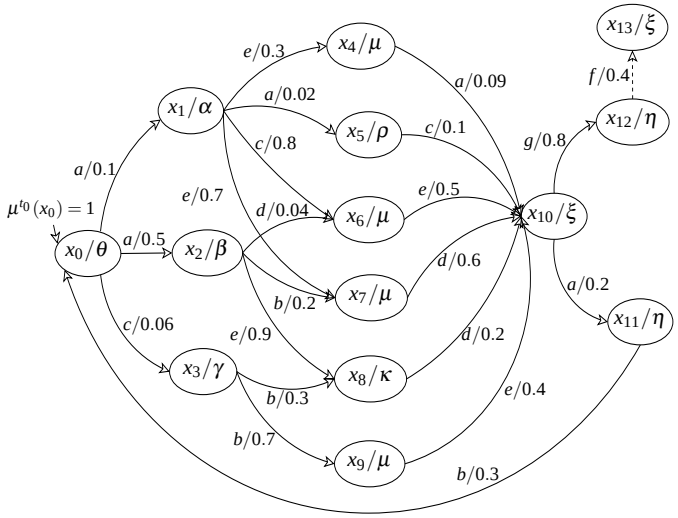


Figure 2. Fuzzy discrete event system model.

- **at time** t_0
  $X_{act}(t_0) = \{(x_0,\mu^{t_0}(x_0)\}$ with $\mu^{t_0}(x_0) = 1,$

$$\begin{cases} X_{succ}(x_0,a_k) = \begin{cases} \{x_1,x_2\} & \text{if } a_k = a,\\ \{x_3\} & \text{if } a_k = c, \end{cases}\\ X_{succ}(x_0,\overset{all}{\rightarrow}) = \{x_1,x_2,x_3\}. \end{cases}$$

  $X_{succ}(x_j,\overset{all}{\rightarrow})$ is the set of all (possible) successors of state x_j,
- **at time** t_1**, input is "a"**
  $X_{act}(t_1) = \{(x_1,\mu^{t_1}(x_1)),(x_2,\mu^{t_1}(x_2))\},$

  and

$$\left.\begin{array}{l} X_{pred}(x_1,t_1) = \\ X_{pred}(x_2,t_1) = \end{array}\right\}\ \{x_0\}$$

and $|X_{pred}(x_1,t_1)|$ is the number of predecessors of state x_1, and

$$|X_{pred}(x_1,t_1)| = |X_{pred}(x_2,t_1)| = 1,$$

and when

$$|X_{pred}(x_j,t)| \leq 1,$$

the state x_j have a single *mv* and Ψ_1 is used to compute $\mu^t(x_j)$ for the state x_j, othewise the function Ψ_2 is used.

The *mv* of x_1 and x_2 is computed by:
$\mu^{t_1}(x_1) = \Psi_1(\mu^{t_0}(x_0), \delta(x_0,a,x_1) = \Psi_1(1,0.1)$,
$\mu^{t_1}(x_2) = \Psi_1(\mu^{t_0}(x_0), \delta(x_0,a,x_2) = \Psi_1(1,0.5)$,

and

$X_{succ}(x_1, \overset{all}{\rightarrow}) = \{x_4,x_5,x_6,x_7\}$,
$X_{succ}(x_2, \overset{all}{\rightarrow}) = \{x_6,x_7,x_8\}$,

- **at time t_2, input is "e"**
  $X_{act}(t_2) = \{(x_4,\mu^{t_2}(x_4)),(x_7,\mu^{t_2}(x_7)),(x_8,\mu^{t_2}(x_8))\}$,

  and

  $\mu^{t_2}(x_4) = \Psi_1(\mu^{t_1}(x_1), \delta(x_1,e,x_4)$,
  $\mu^{t_2}(x_7) = \Psi_1(\mu^{t_1}(x_1), \delta(x_1,e,x_7)$,
  $\mu^{t_2}(x_8) = \Psi_1(\mu^{t_1}(x_2), \delta(x_2,e,x_8)$,

  and

  $X_{succ}(x_4,a) = X_{succ}(x_7,d) = X_{succ}(x_8,d) = \{x_{10}\}$,

  and

  $X_{pred}(x_4,t_2) = X_{pred}(x_7,t_2) = \{x_1\}$,
  $X_{pred}(x_8,t_2) = \{x_2\}$,
  $|X_{pred}(x_4,t_2)| = |X_{pred}(x_7,t_2)| = 1$ and
  $|X_{pred}(x_8,t_2)| = 1$,
- **at time t_3, input is "d"**
  $X_{act}(t_3) = \{x_{10}, \mu^{t_3}(x_{10})\}$,

  and

  $X_{pred}(x_{10},t_3) = \{x_4,x_7,x_8\}$, & $|X_{pred}(x_{10},t_3)| \geq 1$,
  hence, the state x_{10} is forced to take several different *mv*, then Ψ_2 is used to compute $\mu^{t_3}(x_{10})$.

$$\begin{cases} \mu_1(t_3) = \Psi_1(\mu^{t_2}(x_4), \delta(x_4,d,x_{10})), \\ \mu_2(t_3) = \Psi_1(\mu^{t_2}(x_7), \delta(x_7,d,x_{10})), \\ \mu_3(t_3) = \Psi_1(\mu^{t_2}(x_8), \delta(x_8,d,x_{10})), \\ \mu^{t_3}(x_{10}) = \Psi_2[\mu_1(t_3),\mu_2(t_3),\mu_3(t_3)], \end{cases}$$

to compute $\mu^{t_3}(x_{10})$, we can use Maximum multi-membership resolution given by relation (3) or Arithmetic mean multi-membership resolution defined by relation (3).

The fuzzy set of all active output, *i.e.*, output labels together with their *mv*'*s*, at time t denoted as $Y_{act}(t)$, is called the active output set at time t, given by:

$$Y_{act}(t) = \left\{(y_l, \tau^t(y_l)\right\} \text{ and } Y_{act}(\Gamma) = Y_{act}(t_0 + |\Gamma|),$$

where $\tau^t(y_l)$ is the grade membership of the output y_l at time t. In this paper, y_l can be a state with multi-membership. For example,

- at time t_1

$$\begin{aligned} Y_{act}(t_1) &= \left\{(\alpha,\tau^{t_1}(\alpha)),(\beta,\tau^{t_1}(\beta))\right\}, \\ &= \left\{(\alpha,\mu^{t_1}(x_1)),(\beta,\mu^{t_1}(x_2))\right\}, \end{aligned}$$

- at time t_2, the active state x_4 and x_7 generate the same output label μ, *i.e.*, see Figure 4

$$\begin{aligned} Y_{act}(t_2) &= \left\{(\mu,\tau^{t_2}(\mu)),(\kappa,\tau^{t_2}(\kappa))\right\}, \\ &= \left\{(\mu,[\mu^{t_2}(x_4)\ \ \mu^{t_2}(x_7)]),(\kappa,\mu^{t_2}(x_8))\right\}, \end{aligned}$$

most of applications, the output should be crisp. Therefore, the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment and the outputs are assumed to be observable.

A diagnoser must be able to detect and isolates faults and failures (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). In this paper, the diagnoser $D_{\widetilde{G}}$ is a finite-state Automaton which takes the fuzzy output sequence of the system, *i.e.*, $\{(y_1,\tau^{t_1}(y_1),\cdots,(y_k,\tau^{t_k}(y_k))\}$ as its input, and based on this sequence calculates a set $z_k \in 2^X - \{\emptyset\}$ to which $x_i \in X$ must belong a time that pair $(y_k,\tau^{t_k}(y_k))$ was generated. The diagnoser $D_{\widetilde{G}}$ is given by:

$$D_{\widetilde{G}} = (Z,Y,\zeta,\lambda,z_0,\Omega),$$

with

- Z is the set of standard diagnoser state,
- Y is the set of standard diagnoser input,
  we recall, Y is the output of model $\widetilde{G}$,
- λ is the set of standard dianoser output,
- $\zeta : Z \times Y \times \rightarrow Z \times \lambda$ is the standard diagnoser state transition function,
- z_0 is the start state set of the standard diagnoser,
- $\Omega \in Z$ is the (non-empty) set of terminal states

Let ζ_1 and ζ_2 be the two projections of ζ of $D_{\widetilde{G}}$, with ζ_1 and ζ_2 are given by

$$\begin{cases} \zeta_1(z_k,y_{k+1}) = \{z_{k+1} \mid \exists\ \lambda_i \wedge (z_{k+1},\lambda_i) \in \zeta(z_k,y_{k+1})\}, \\ \zeta_2(z_k,y_{k+1}) = \{\lambda_i \mid \exists\ z_{k+1} \wedge (z_{k+1},\lambda_i) \in \zeta(z_k,y_{k+1})\}, \\ \zeta(z_k,y_{k+1}) = (\zeta_1(z_k,y_{k+1}),\zeta_2(z_k,y_{k+1})). \end{cases}$$

with $\lambda_i = \lambda(z_{k+1})$ and $z_k \subseteq Z$ is the state estimate of $D_{\widetilde{G}}$ at time k.

The diagnoser state transition is given by

$$\begin{cases} (z_{k+1}, \lambda(z_{k+1})) = \zeta(z_k, y_{k+1}), \\ \lambda(z_{k+1}) = \zeta_2(z_k, y_{k+1}), \\ z_{k+1} = \zeta_1(z_k, y_{k+1}), \\ \quad = X_{succ}(z_k, \overset{all}{\rightarrow}) \cap \zeta_1(z_k, y_{k+1}), \end{cases}$$

Figure 5 shows the standard diagnoser for the discrete event system model of Figure 4, with $z_0 = \{x_0\}$. Each state of the diagnoser $D_{\widetilde{G}}$, shown as a rounded box in Figure 5, is a set of states of the system. An output symbol and a failure condition are associated with each diagnoser state. For instance, to see the importance of having a complete model for the diagnoser, we suppose at time k the output sequence $''\theta\alpha\mu\xi\eta''$ is observed, then the state estimate is $z_{10} = \{x_{11}, x_{12}\}$ and systems condition from z_0 is $\lambda(z_{10}) = N$. The successors of state estimate z_{10} is: $Z_{succ}(z_{10}) = z_{11} = \{x_{13}\}$ or $Z_{succ}(z_{10}) = z_0 = \{x_0\}$. If the next output symbol y_{k+1} is anything other than ξ or θ, we get

$$Z_{succ}(z_{10}) = X_{succ}(z_1, \overset{all}{\rightarrow}) \cap \zeta_1(z_1, y_{k+1}) = \emptyset,$$

that means the observation generated after y_k is inconsistent with the model dynamic and the diagnoser cannot proceed. When the output sequence is inconsistent with the model of the system, then we have to revise the model of $\widetilde{G}$ by adding new state(s) and/or new transition(s) respectively in X and Σ, that we believe are missing in the nominal model. This situation may be interpreted as a normal or abnormal situation, because we add new states and/or transitions. Detecting and adding new states and/or transitions in X and/or in Σ of $\widetilde{G}$ is called learning diagnoser. A algorithm of a learning diagnoser is presented in the next section.

5. A ALGORITHM OF A LEARNING DIAGNOSER

A learning diagnoser is a standard diagnosis that tolerant of missing information, *i.e.*, transitions and states, about the system to be diagnosed. The learning diagnoser must be able to learn the true model of the system $\widetilde{G}$, when missing information about the system are presented.

Let a_{new} be a new event detected and not found in Σ of system $\widetilde{G}$, then the new set of input events of $\widetilde{G}$ is given by

$$\Sigma_{new} = \Sigma \cup \{a_{new}\}.$$

A transition $x_d \xrightarrow{a_{new}} x_a$ is ordered pair of state denoting a transition from the state x_d to the state x_a. Let φ' be the extend function transition of φ of the system $\widetilde{G}$ such that

$$\varphi_{new}(x_d, a_i) = \begin{cases} x_a \text{ if } a_i = a_{new} \ \& \ \begin{cases} \Sigma \leftarrow a_{new}, \\ \text{and} \\ X \leftarrow x_a \quad \text{if} \quad x_a \notin X, \end{cases} \\ \varphi_1(x_d, a_i) \text{ otherwise,} \end{cases}$$

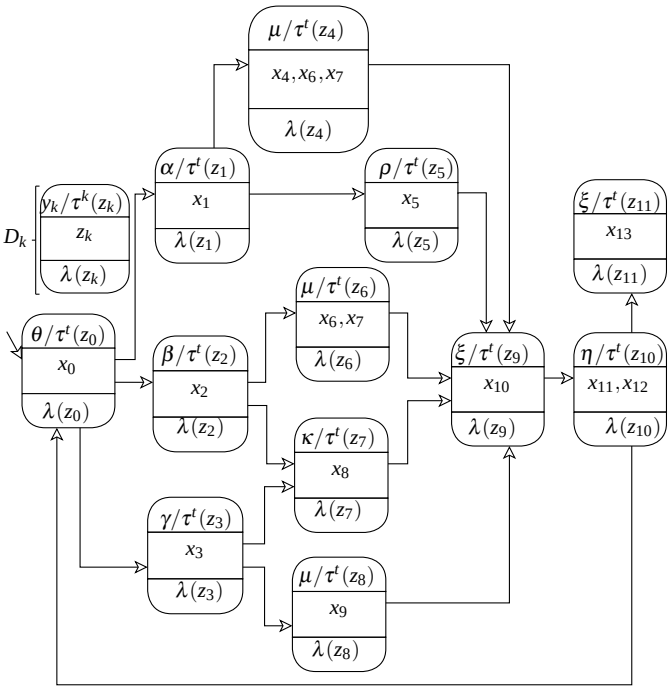


Figure 3. Diagnoser of fuzzy discrete event system model shown in Figure 4, $\lambda(z_i)_{i=0\,to\,10} = N$ and $\lambda(z_{11}) = F_1$.

Let be a dynamic model $\widetilde{G}'$ of $\widetilde{G}$ defines as

$$\widetilde{G}' = extend(\widetilde{G}, X', \Pi) = (X \cup X', \Sigma \cup \Pi, Y, \varphi_{new}, \widetilde{x}_0).$$

And $\widetilde{G}'$ is called the extension of $\widetilde{G}$ by X' and Π, with X' is the set containing all new states and Π is the set containing all new transitions founded. The set transition Π is empty, if the model G of the system is consistent with the output sequence.

The algorithm presented in Algorithm 1 is the algorithm for the learning diagnoser and evolving fuzzy state automaton.

6. APPLICATION EXAMPLE

Nowadays, the crisis management is an important challenge for medical service and research, to develop new technical of decision support system to guide the decision makers. The crisis management is a special type of collaboration, therefore several aspects must be considered. The more important aspect in a crisis management is the coordination (and communication) between different actors and groups involved in the crisis management. Hence, the capacity to take fast and efficient decisions is a very important challenge for a better exit of crisis. Because the context and characteristics of crisis such as extent of actors and roles, the management becomes more difficult in order to take decisions, but also to exchange information or to coordinate different groups involved. The difficult to take a decision can be also due to random factors, such as stress, emotional impact, road conditions, weather conditions, etc. During the crisis management, it is hard to

initialization;
while *input is* a_k *and active state time* $t-1$ *is* x_i **do**
  read symbol a_k;
  $x_j = \varphi_1(x_i, a_k)$;
  $y_j = y_{k+1} = \varphi_2(x_i, a_k)$;
  $X_{succ}(x_i, a_k) = \{\forall\, x_s \in X \mid x_s \in \varphi_1(x_i, a_k)\}$;
  if x_i *is the start state and time is* t_0 **then**
    $X_{pred}(x_j, t) = \emptyset$;
  else
    $X_{pred}(x_j, t) = \{\forall\, x_i \in X \mid x_j \in \varphi_1(x_i, a_k)\}$;
  end
  if $(X_{succ}(x_i, a_k) \cap \zeta_1(z_k, y_{k+1}) \neq \emptyset)$ **then**
    if $(|X_{pred}(x_j, t)| = \emptyset)$ **then**
      $X_{act} = x_0$;
      $X_{succ}(x_0, a_k) = \{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\}$;
    else if $(|X_{pred}(x_j, t)| = 1)$ **then**
      single mv of all active states ;
      $\mu^t(x)$ *of each state* $x \in X_{act}$ *is computed by* ;
      $\mu^t(x_j) = \Psi_1(\mu^t(x_j), \delta(x_i, a_k, x_j))$;
      $X_{act} = \{(x_j, \mu^t(x_j)\}$;
      $X_{succ}(x_j, a_k) = \{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\}$;
    else
      active state have been forced to take different several mv ;
      $m = |X_{pred}(x_j, t)|$;
      for $i = 1$ *to* m **do**
        $\mu_i = \Psi_1(\mu^{t-1}(x_i), \delta(x_i, a_k, x_j))$;
      end
      $\mu^t(x_j) = Max(\mu_1, \mu_2, \cdots, \mu_{m-1}, \mu_m)$;
      $X_{act} = \{(x_j, \mu^t(x_j)\}$;
      $X_{succ}(x_j, a_k) = \{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\}$;
    end
    Diagnoser method ;
    go to D_k;
  else
    go to inconsistency;
    detection of new transition and/or state;
    $X_{succ}(x_i, a_k) \cap \zeta_1(z_k, y_{k+1}) = \emptyset$;
    we suppose for all new transition;
    $\delta(x_i, a_k, x_j) = 0$;
    if $(x_j \in X \& a_k \in \Sigma)$ **then**
      new transition between x_i*(past state) to* x_j *(active state)* ;
    else if $x_j \in X$ & $a_k \notin \Sigma$ **then**
      update Σ;
      $\Sigma \leftarrow a_k$;
    else
      update X *and* Σ;
      $X \leftarrow x_j$;
      $\Sigma \leftarrow a_k$;
    end
  end
end

Algorithm 1: Evolving fuzzy finite state automaton

say exactly an actor's stress has changed from low to high. For this reason, it is important to integrate these factors in the model of crisis management for decision-making. The *FFA* presented above is used to takes into account the stress of the actors involved in the crisis management.

6.1. Our *FFA* model of crisis management

In this paper, we propose a model (no generic model) applied on the team *SAMU* [1] from Hospital of Troyes in France, during *TEAN* [2] exercise.

The team of *SAMU* is composed of the following actors:

- Rear Base [3] (*RB*): Operations Coordination,
- Communication Center (*CC*): collecting information and sharing with *RB*,
- First Team: first intervention, sending the first evaluation (result) about the crisis to the *CC*,
- Advanced Medical Post (*AMP*): Intervention and evacuation of victims, sending the complete evaluation to the *CC*.

The *FSA* of the *TEAN* exercise is shown in Figure 4.

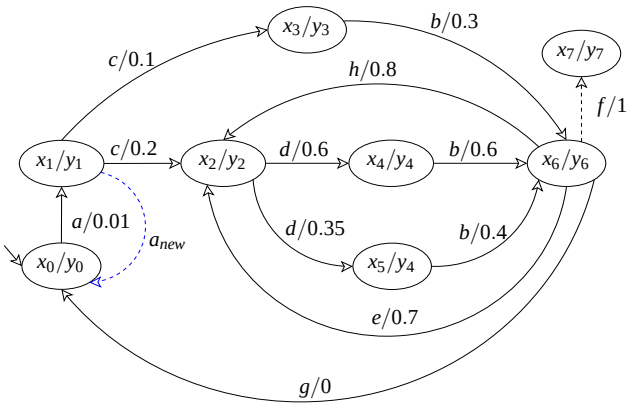


Figure 4. A example of modelisation of a scenario of crisis with finite state automaton and the weight corresponds to the stress of actors involved.

The discrete event model showed in Figure 4 for *TEAN* exercise, allows one hand to monitor the communication and coordination between various groups involved in crisis management, and also to supervise some specific behaviors that are critical situations. Thus the factor's stress of the actors involved is estimated for decision-making.

Consider the *FFA* in Figure 4 with several transition overlaps and several output labels. It is specified as:

$$\widetilde{G}_n = (X, \Sigma, \delta, Y, \widetilde{x}_0, F),$$

[1] SAMU is Service Emergency Medical Assistance.
[2] TEAN is the name of the exercise.
[3] Other word, Rear Base is decision makers

The dashed line in Figure 4, between states 6 and 7 represents a critical event. The occurrence of event $''f''$ bring the system in or critical mode corresponding to state x_7 and $\omega_{i,j}$ is the stress of actors involved in crisis management.

In this example

$X = \{x_0, x_1, \cdots, x_7\}$, is the set of states, which occur with different, membership degree $(\mu^t(x_0), \cdots, \mu^t(x_7))$.
$\Sigma = \{a, b, c, d, e, f, g, h\}$, set of input symbols,
$Y = \{y_1, y_2, y_3, y_4, y_6, y_7\}$, set of output events,
$\widetilde{x_0} = \{(x_0, \mu^{t_0}(x_0) = 0)\}$, starting state,

$$\lambda(x_i) = \begin{cases} F_1(\text{abnormal mode}), & \text{if i=7,} \\ N(\text{normal mode}), & \text{otherwise.} \end{cases}$$

Table 1. List and definition of the states.

States	Definition
x_0	No crisis
x_1	Onset Crisis
x_2	Information received at the communication center (CC)
x_3	Information arrived at the police center
x_4	Information received at the Emergency department
x_5	Information arrived at the Advanced Medical Post (AMP)
x_6	Information received at the accident area
x_7	The model is unpredictable for this crisis situation

Table 2. List and definition of outputs.

Output labels	Definition
y_0	No coming call
y_1	Accident is happen
y_2	Information arrived to CC
y_3	Information arrived to police office
y_4	Preparation of the Intervention Team
y_5	Preparation of the AMP
y_6	New Actors arrived in the accident area
y_7	uncontrolled situations (conditions)

Table 3. List and definition of the transitions (events).

events	Definition
a	A call from (or about) a accident
b	Sending Team to the accident site
c	Sending information to CC and police office
d	Sending information to Emergency
e	Sending the first evaluation to CC
h	Sending final evaluation to CC
f	End of crisis management without success
g	End of crisis management with success

In this example, we suppose at the beginning $\mu^{t_0}(x_0) = 0$ (i.e, stress level is very low) and all the other *mv* are computed by using approaches presented in section 3.

Assuming that $\widetilde{G}_n$ starts operating at time t_0 and the next three input are "**a**" respectively (one at a time), active states and their $mv's$ at each time step are as follows.

- **at time** t_0
  $X_{act}(t_0) = \{(x_0, \mu^{t_0}(x_0)\}$

  $$\begin{cases} X_{succ}(x_0, a) = x_1, \\ X_{succ}(x_0, \overset{all}{\rightarrow}) = \{x_1\}. \end{cases}$$

  $X_{succ}(x_j, \overset{all}{\rightarrow})$ is the set of all successors of state x_j,
- **at time** t_1**, input is "a"**
  $X_{act}(t_1) = \{(x_1, \mu^{t_1}(x_1))\}$,
  $Y_{act}(t_1) = \{(y_1, \tau^{t_1}(z_1))\}$, and $\tau^{t_1}(z_1) = \tau^{t_1}(x_1) = \mu^{t_1}(x_1)$ at time t_1 the weight corresponding to the stress of the people involved is $\omega_{0,1} = 0.01$ and this weight is estimated by the expert of the crisis management.
  $X_{pred}(x_1, t_1) = x_0$, and $|X_{pred}(x_1, t_1)|$ is the number of predecessors of active state x_1. $|X_{pred}(x_j, t)| = 1$, then, the active state x_1 is not forced to take multi-membership.
  $X_{succ}(x_1, c) = \{x_2, x_3\}$,

6.2. Diagnoser model of $TEAN$ exercise

The standard diagnoser for the fuzzy discrete event system of crisis management model illustrated in Figure 4 is shown in Figure 5, with $z_0 = \{x_0\}$. Each state of the diagnoser $D_{\widetilde{G}_n}$, shown as a rounded box in Figure 5, is a set of states of the system. An output symbol corresponding to the operating condition of the system is associated with each diagnoser state. For example, to see the importance of having a complete model for the diagnoser, we suppose at time t_1 the output sequence $''y_0 y_1''$ (*see* Figure 4) is observed, then the state estimate is $z_1 = \{x_1\}$ and the operating condition from z_0 is $\lambda(z_1) = N$. The successors of state estimate z_1 is: $Z_{succ}(z_1) = \{z_2, z_3\} = \{x_2, x_3\}$. If the next output symbol y_{t+1} is y_0, we get

$$Z_{succ}(z_1) = X_{succ}(z_1, \overset{all}{\rightarrow}) \cap \zeta_1(z_1, y_{t+1}) = \emptyset,$$

that means the observation generated after y_1 is inconsistent with the model dynamic and the diagnoser cannot proceed. When the output sequence is inconsistent with the system's model, then we have to revise the model of $\widetilde{G}_n$ by adding in this application a new transition (a_{new}) from the state x_1 to the state x_0(s) (see Figure 4). This situation may be interpreted as a normal or abnormal situation. Detecting and adding new states and/or transitions in X and/or in Σ of $\widetilde{G}$ is called learning diagnoser.

7. CONCLUSION

In this paper, we have dealt with the failure diagnosis of fuzzy finite state automaton for systems operating in non-stationary environment. We have presented in our paper, the definition

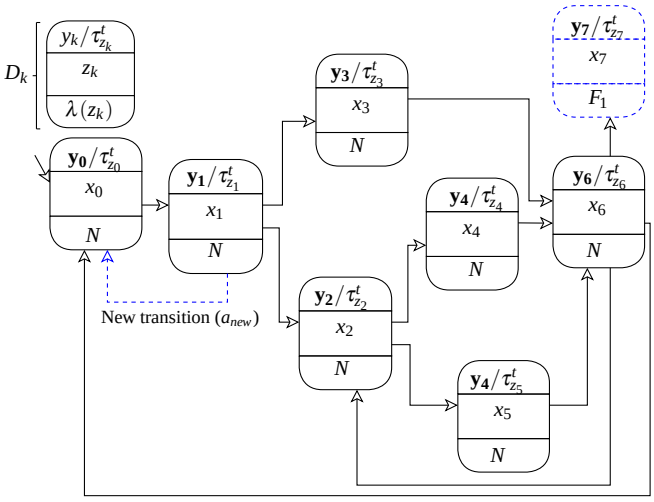


Figure 5. Diagnoser of fuzzy discrete event system model shown in Figure 4.

of a crisp discrete event system and fuzzy discrete event system. The main advantage of fuzzy finite state automaton, to handle imprecise and uncertain data is presented. We have formalized the construction of the learning diagnoser based on evolving fuzzy finite state automaton that are used to perform fuzzy diagnosis. In particular, we have propose a algorithm for learning diagnoser based on evolving fuzzy finite state automaton that allows to add new transitions and states. The newly proposed diagnoser approach allows us to deal with the problem of failure diagnosis for fuzzy discrete event system, which many better deal with the problem of fuzziness, impreciseness and uncertainness in the failure diagnosis.

The potential application of learning diagnosis based on fuzzy finite state automaton is in solving the ambiguity in a fault diagnosis problem especially in the case of multiple faults.

Future work will focus on the proposal of fuzzy states of crisis management by using fuzzy finite automaton that takes into account of a random vector as such the stress, weather condition and emotional impact of the actors involved in crisis management.

ACKNOWLEDGMENT

This work is supported by the CPER prject AidCrisis, sponsored by the Champagne-Ardenne region and the French ministry of higher education and research.

REFERENCES

Cabasino, M. P., & Alessandro Giua, C. S. (2010). Fault detection for discrete event systems using petri nets with unobservable transition. *Automatica, vol. 46*, pp. 1531-1539.

Cao, Y., & Ying, M. (2005). Supervisory control of fuzzy discrete event systems. *IEEE transactions on systems, Man, and Cybernetics-part B: Cybernetics, vol. 35 (2)*, pp. 366-370.

Cao, Y., & Ying, M. (2006). Observability and decentralized control of fuzzy discrete-event systems. *IEEE Trans. Fuzzy Syst., vol. 14*, pp. 202-216.

Cassandras, C. G., & Lafortune, S. (1999). Introduction to discrete event systems. *Boston, MA:Kluwer*.

Doostfatemeh, M., & Kremer, S. C. (2004). The significance of output mapping in fuzzy automato. *Proceedings of the 21th Iranian Conference on Electrical Engineering (ICEE)*.

Doostfatemeh, M., & Kremer, S. C. (2005). New directions in fuzzy automaton. *International Journal of Approximate Reasoning, vol. 38*, pp. 175-214.

Dzelme-Berzina, I. (2009). Mathematical logic and quantum finite state automata. *Theoretical Computer Science, vol. 410*, pp. 1952-1959.

Gerasimos, G. R. (2009). Fault detection and isolation based on fuzzy automaton. *Information Sciences, vol. 179*, pp. 1893-1902.

Kwong, R. H., & Yonge-Mallo, D. L. (2011). Fault diagnosis in discrete-event systems: Incomplete models and learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 41 (1)*, pp. 118 - 130.

Liu, F., & Qiu, D. (2009a). Diagnosability of fuzzy discrete-event system: A fuzzy approach. *IEE Transactions on Fuzzy Systems, vol. 17 (2)*, pp. 372-384.

Liu, F., & Qiu, D. (2009b). Diagnosability of fuzzy discrete event systems: A fuzzy approach. *IEEE Transactions on fuzzy system, vol. 17 (2)*, pp. 372-384.

Luo, M., Li, Y., Sun, F., & Liu, H. (2012). A new algorithm for testing diagnosability of fuzzy discrete event systems. *Information Sciences, vol. 185*, pp. 100 - 113.

Moamar, S.-M., & Billaudel, P. (2012). Abrupt and drift-like fault diagnosis of concurent discrete event systems. *Machine Learning and Applications (ICMLA), vol. 2*, pp. 434-439.

Moghari, S., Zahedi, M. M., & Ameri, R. (2011). New direction in fuzzy tree automata. *Iranian Journal of Fuzzy Systems, vol. 8 (5)*, pp. 59-68.

Mukherjee, K., & Ray, A. (2014). Statesplittingandmerging-inprobabilisticfinitestate automataforsignalrepresentationandanalysis. *Signal Processing, vol. 104*, pp. 105-119.

Patela, A. M., & Joshi, A. Y. (2013). Modeling and analysis of a manufacturing system with deadlocks to generate the reachability tree using petri net system. *International Conference On DESIGN AND MANUFACTURING, IConDM 2013, vol. 64*, pp. 775-784.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of dis-

crete event systems. *IEEE Transaction On Automatic Contol, vol. 40 (9)*, pp. 1555-1575.

Sardouk, A., Mansouri, M., Merghem-Boulahia, L., & Gaiti, D. (2013). Crisis management using mas-based wireless sensor networks. *Computer Networks, vol. 57*, pp. 29-45.

Thomas, W. (1990). Handbook of theoretical computer science. *Elsevier, B.*

Traore, M., Moamar, S.-M., & Billaudel, P. (2013). Learning diagnoser and supervision pattern in discrete event system : Application to crisis management. *Annual Conference of the Prognostics and Health Management Society, ISBN-978-1-936263-06-6, New Orleans, USA*, 694-701.

BIOGRAPHIES

Dr. Moussa Traore is a Postdoctoral fellow at the University of Technology of Troyes, France working under supervision of prof. Eric Chatelet. He is working on Fault Diagnosis in discrete event systems and fuzzy discrete event systems. He is working on an AidCrisis project sponsored by Champagne-Ardenne region and the French ministry of higher education and research. He obtained his *Ph.D* degree (Control System and Signal Processing) from University Lille 1 in France. He received his Master of Science (Optimization and safety of functioning Systems) from University of Technology of Troyes, France, 2006 and Bachelor's degree in Electrical Engineering from Faculty of Sciences and Technology of Nouakchott, Mauritania, 2004. His research area is diagnostic and prognostic of dynamic systems in non-stationary environment, for continuous and discrete systems. His research work allowed him to publish 4 journal papers, 7 papers with proceedings at international conferences and 3 papers with proceedings at national (France) and international conferences (in French) in the area of diagnosis, prognosis and predictive maintenance. He served as International reviewer for several International Conferences.

Dr. Hossam A. Gabbar Dr. Hossam A. Gabbar is Associate Professor in the Faculty of Energy Systems and Nuclear Science, and cross appointed in the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology (UOIT). He obtained his Ph.D. degree (Safety Engineering) from Okayama University (Japan), while his undergrad degree (B.Sc.) is in the area of automatic control from Alexandria University, Egypt. He is specialized in safety and control engineering where he worked in process control and safety in research and industrial projects in Japan and Canada. Since 2004, he was tenured Associate Professor in the Division of Industrial Innovation Sciences at Okayama University, Japan. And from 2001, he joined Tokyo Institute of Technology and Japan Chemical Innovative Institute (JCII), where he participated in national projects related to advanced distributed control and safety design and operation synthesis for green energy and production systems. He developed new methods for automated control recipe synthesis and verification, safety design, and quantitative and qualitative fault simulation.

He is a Senior Member of IEEE, the founder of SMC Chapter - Hiroshima Section, the founder and chair of the technical committee on Intelligent Green Production Systems (IGPS), and Editor-in-chief of International Journal of Process Systems Engineering (IJPSE), president of RAMS Society, and editorial board of the technical committee on System of Systems and Soft Computing (IEEE SMCS). He is invited speaker in several Universities and international events, and PC / chair / co-chair of several international conferences. Dr. Gabbar is the author of more than 110 publications, including books, book chapters, patent, and papers in the area of safety and control engineering for green energy and production systems. His recent work is in the area of risk-based safety and control design for energy conservation and supply management, and smart grid modeling and planning with distributed generation.

Health Monitoring of Hybrid Systems Using Hybrid Particle Petri Nets

Quentin Gaudel[1], Elodie Chanthery[2], and Pauline Ribot[3]

[1,2,3] *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*
quentin.gaudel@laas.fr, elodie.chanthery@laas.fr, pauline.ribot@laas.fr

[1,2] *Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France*

[3] *Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France*

ABSTRACT

This paper presents an approach of model-based diagnosis for the health monitoring of hybrid systems. These systems have both continuous and discrete dynamics. Modified Particle Petri Nets, initially defined in the context of hybrid systems mission monitoring, are extended to estimate the health state of hybrid systems. This formalism takes into account both uncertainties about the system knowledge and about diagnosis results. The generation of a diagnoser is proposed to track online the system health state under uncertainties by using particle filter. To include more complex characteristics of the system, as its degradations for prognosis purpose, an enriched formalism called Hybrid Particle Petri Nets is defined.

1. INTRODUCTION

Systems have become so complex that it is often impossible for humans to capture and explain their behaviors as a whole, especially when they are exposed to failures. It is therefore necessary to develop tools that can support operator tasks but that also reduce the global costs due to unavailability and repair actions. An efficient diagnosis technique has to be adopted to detect and isolate faults leading to failures. Diagnosis uses a behavioral model of the system and online observations to determine the behavioral state of the system. Uncertainties in diagnosis can be taken into account by giving as much information as possible about the ambiguous state likelihood. On the other side, systems are continuously degrading depending on operational conditions. Knowing available information on the system, it is possible to establish physical degradation laws or time-dependent fault probability distributions based on the feedback. It is then interesting to take into account this temporal and/or stochastic information about the system degradation. Health monitoring consists in evaluating the current health state of the system through a diagnosis and a degradation law value. The health state is represented by a degradation measure for the system in a specific behavioral state (Vinson, Ribot, Prado, & Combacau, 2013). Its estimation is the first step to perform later prognosis and to compute the remaining useful life (RUL) of the system. A formal generic modeling framework for health monitoring of complex heterogeneous systems has been presented in (Ribot, Pencolé, & Combacau, 2013) and encapsulates knowledge about the system behavior and degradation used by diagnosis and prognosis. Uncertainties in the model and diagnosis results are taken into account by estimating interval ranks for parameters. Another common framework for diagnosis and prognosis has been proposed in (Roychoudhury & Daigle, 2011). This article presents a state model that specifies the nominal behavior of the system and fault progression over time. However, it only represents systems with a continuous dynamics without discrete or hybrid aspect.

Recent industrial systems exhibit an increasing complexity of dynamics that are both continuous and discrete. It has become difficult to ignore the fact that most systems are hybrid (Henzinger, 1996). In previous works (Chanthery & Ribot, 2013), we extended the diagnosis approach proposed in (Bayoudh, Travé-Massuyes, & Olive, 2008) in order to integrate diagnosis and prognosis for hybrid systems. The approach uses hybrid automata and stochastic models for the system degradation. The main drawback of this approach is that the discrete event system oriented diagnosis framework explodes in number of states and it does not seem the best suited for the incorporation of the highly probabilistic prognosis task. To have a more compact representation and to capture all uncertainties related to the system, to the observations and to the diagnosis results, we propose to consider the

Quentin Gaudel et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

formalism of Modified Particle Petri Nets (MPPN) defined in (Zouaghi, Alexopoulos, Wagner, & Badreddin, 2011a). Moreover this representation is intuitive and facilitates the modeling task. MPPN are an extension of Particle Petri nets (Lesire & Tessier, 2005) that combines a discrete event model (Petri net) and a continuous model (differential equations). The main advantage of MPPN is that uncertainties and hybrid dynamics are taken into account. Particle filter is used to integrate probabilities in the continuous state estimation process. MPPN have been used for supervision and planning, but never for health monitoring, diagnosis and/or prognosis.

MPPN representation is useful in capturing all uncertainties about the state knowledge and about the observations. As wide as can be the range of feature representations provided by MPPN, we did not succeed in modelling a characteristic that depends on a discrete state and a continuous state of the system. That is why we propose to define what we call Hybrid Particle Petri Nets (HPPN) in order to model both behavior and degradation of hybrid systems in the context of health monitoring. The HPPN formalism enriches MPPN to model available knowledge about hybrid characteristics of the system. The paper is organized as follows. Section 2 recalls the MPPN framework and presents how it can be used for behavioral health monitoring. In Section 3 a hybrid diagnosis technique is proposed based on the generation of a behavioral diagnoser using the MPPN formalism. The MPPN enrichment is defined in Section 4 as Hybrid Particle Petri Nets to take the system degradation into account by interacting with the hybrid behavioral model. Some conclusions and future work are discussed in the final section.

2. MODIFIED PARTICLE PETRI NETS FOR MONITORING

In this section, the Modified Particle Petri Nets (MPPN) formalism is described according to the work of (Zouaghi et al., 2011a). First the model structure and its online process are detailed and then a way to use it to represent system health model is presented.

2.1. Definition

Modified Particle Petri Nets are defined as a tuple $< P, T, Pre, Post, X, C, \gamma, \Omega, M_0 >$ where:

- P is the set of places, partitioned into numerical places P^N and symbolic places P^S.
- T is the set of transitions (numerical T^N, symbolic T^S and mixed T^M).
- Pre and $Post$ are the incidence matrices of the net, of dimension $|P| \times |T|$.
- $X \subset \Re^n$ is the state space of the numerical state vector.
- C is the set of dynamics equations of the system associated with numerical places, representing continuous state evolution.
- $\gamma(p^S)$ is the application that associates tokens with each symbolic places $p^S \in P^S$.
- Ω is the set of conditions associated with the transitions (numerical Ω^N and symbolic Ω^S).
- M_0 is the initial marking of the net.

MPPN can model system behaviors. A basic example of a system behavior modeled with MPPN is illustrated in Figure 1.

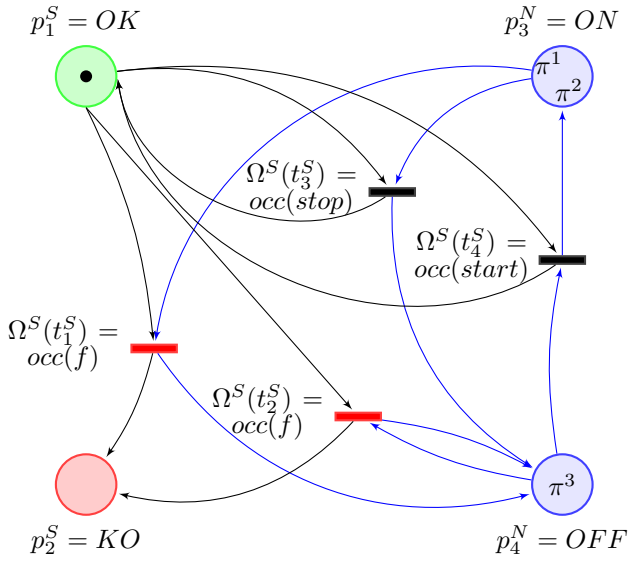


Figure 1. Example of MPPN.

There are four places in this MPPN: $P = \{p_1^S, p_2^S, p_3^N, p_4^N\}$. Two symbolic places p_1^S and p_2^S represent the two discrete modes of the system, respectively when the system is working well (OK) and when the system has failed (KO). Two numerical places p_3^N and p_4^N represent the two continuous behaviors of the system, respectively when it is turned on (ON) and when it is turned off (OFF). There are four symbolic transitions is this MPPN: $T = \{t_1^S, t_2^S, t_3^S, t_4^S\}$. They represent occurences of discrete events. t_1^S and t_2^S represent the occurence of a fault event f respectively when the system is turned on and turned off and let the system go from the OK mode to the KO mode. t_3^S represents the occurence of a mission event $stop$ that turns off the system when it is turned on and is in OK mode. Finally, t_4^S represents the occurence of a mission event $start$ that turns on the system when it is turned off and is in OK mode.

A numerical place $p^N \in P^N$ is associated with a set of dynamics equations representing the continuous behavior of the system. Numerical places thus model continuous dynamics of the system. Numerical places are marked by a set of particles $\pi_k^i = [x_k^i, w_k^i]$ with $i \in \{1, ..., |M_k^N|\}$ where M_k^N is the set of all the particles in the net at time k. Particles are defined by their corresponding numerical state vector $x_k^i \in X$

and their weight $w_k^i \in [0,1]$ at time k. The set of particles represents an uncertain distribution over the value of the numerical state vector.

Symbolic places model the behavioral modes of the system. A symbolic place $p^S \in P^S$ is marked by configurations δ_k^j with $j \in \{1,...,|M_k^S|\}$ where M_k^S is the set of configurations in the net at time k. The set of configurations represents all the possible current modes of the system.

The marking of the net is composed of tokens, that can be numerical tokens (particles) or symbolic tokens (configurations). The marking M_k of the MPPN at time k consists of both kinds of tokens:

$$M_k = \{M_k^S, M_k^N\} \quad (1)$$

For example, in Figure 1, the numerical place p_3^N is marked by the set of particles $\{\pi^1, \pi^2\}$ and the symbolic place p_1^S is marked by a configuration. The marking of the illustrated MPPN is then $M_k = \{[0,1]', [\{\pi^1, \pi^2\}, \{\pi^3\}]'\}$.

A transition models a change in the continuous dynamic and/or a change of the system mode. A symbolic transition is conditioned by an observable discrete event. A numerical transition is conditioned by a set of constraints on continuous observable variables. Finally, a mixed transition is conditioned by an observable discrete event and a set of constraints on continuous observable variables.

2.2. Firing Rules

This section recalls the basic ideas of MPPN firing rules. More formal details about the firing rules of the different transitions can be found in (Gaudel, Chanthery, Ribot, & Le Corronc, 2014). A numerical transition $t_j^N \in T^N$ is associated with conditions $\Omega^N(t_j^N)$, where $\Omega^N(t_j^N)(\pi) = 1$ if the particle satisfies the conditions. For example, if $\pi = [x, \mathrm{w}]$ follows the constraint equation c and b is a trigger value, a numerical condition can be defined as $\Omega^N(t_j^N)(\pi) = (c(x) > b)$. $\Omega^S(t_j^S) = occ(e)$ represents the conditions assigned to a symbolic transition $t_j^S \in T^S$. $occ(e)$ is a boolean indicator of the occurrence of the discrete event e : $occ(e) = 1$ if e has occurred. Then, a configuration δ satisfies the condition $\Omega^S(t_j^S)$ when $\Omega^S(t_j^S)(\delta) = 1$, ie. when the event e has occurred.

The numerical firing uses the concept of classical firing with the particles satisfying the numerical condition and the concept of pseudo-firing (ie. duplication) for the configurations. The duplication of configurations represents uncertainty about the occurrence of an unobservable discrete event. An example of a numerical firing from marking at time k to marking at time $k+1$ is illustrated in Figure 2(a). In this example, t_1^N only has a numerical condition because it is a numerical transition. Particle π^3 satisfies the numerical condition $\Omega^N(t_1^N)$ and thus is moved through the transition t_1^N to p_4^N. The configuration in place p_1^S is duplicated in p_2^S.

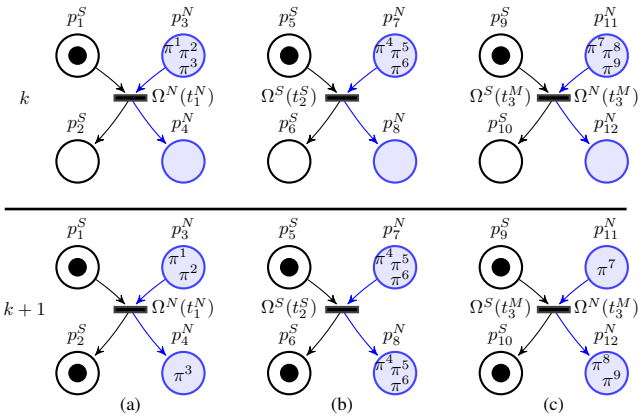


Figure 2. Illustration of firing rules of numerical (a), symbolic (b) and hybrid (c) transitions.

The symbolic firing uses the concept of pseudo-firing for particles and configurations. The pseudo-firing of all the tokens models uncertainty about the occurence and the non occurrence of an observable discrete event. Figure 2(b) illustrates an example of a symbolic firing. The symbolic transition t_2^S only has a symbolic condition. No token satisfies the condition $\Omega^S(t_2^S)$, however all tokens are duplicated.

Mixed transitions are introduced in (Zouaghi et al., 2011a) to model the interaction between discrete events and system continuous dynamics. In the referred article, they were called "hybrid transitions". A mixed transition merges a symbolic transition with a numerical transition to correlate discrete observations with continuous observations. The firing of the symbolic transition only depends on a discrete event, but the simultaneous firing of the numerical transition models the dependency of the mixed transition on the symbolic part because discrete events are part of the process behavior. A mixed transition $t_j^M \in T^M$ is then associated with both numerical conditions $\Omega^N(t_j^M)$ and symbolic conditions $\Omega^S(t_j^M)$.

The mixed firing uses the concept of classical firing with the particles satisfying the numerical condition and the concept of pseudo-firing with the configurations satisfying the symbolic condition. The pseudo-firing of configurations models uncertainty about the occurrence of an observable discrete event which is supported by a change of continuous dynamics. An example of a mixed firing is illustrated in Figure 2(c). t_3^M is a mixed transition therefore it has a symbolic condition and a numerical condition. The configuration in place p_9^S is duplicated because it satisfies the symbolic condition $\Omega^S(t_3^M)$. Regarding the numerical part, particles π^8 and π^9 satisfy $\Omega^N(t_3^M)$ and so they are moved through t_3^M. Furthermore, π^7 stays in place p_{11}^N because it does not satisfy $\Omega^N(t_3^M)$.

Heterogeneous systems are defined as systems that have a discrete, continuous or both discrete and continuous dynamics.

MPPN can easily model heterogeneous systems by using only the symbolic or numerical subpart of the model or both in the case of hybrid systems.

2.3. State Estimation

The problem of hybrid state estimation in MPPN has been introduced in (Zouaghi et al., 2011a) and consists of a prediction step and a correction step, illustrated in Figure 3.

For the sake of clarity in this paper we assume that a hybrid state is represented by a couple (p_i^S, p_j^N) of a symbolic place and a numerical place. The initial marking of the MPPN is $M_0 = \{M_0^S, M_0^N\}$ and the estimated marking at time k is $\hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N\}$ where $\hat{M}_k = \hat{M}_{k|k}$. The observations start at time $k = 1$, $O_1 = (O_1^S, O_1^N)$ where O^S and O^N respectively represent the observations corresponding to the symbolic part and the numerical part.

(1) The prediction is based on the evolution of the MPPN marking and on the estimation of the particle values. It aims at determining all possible next states of the system $\hat{M}_{k+1|k} = \{\hat{M}_{k+1|k}^S, \hat{M}_{k+1|k}^N\}$. A noise is added during the particle values update to take into account uncertainty about the dynamics equations and thus about the continuous system model.

(2) The correction is based on the update of the prediction according to new observations on the system.

- **(a)** A numerical correction, based on particle filter algorithms, produces a probability distribution Pr_{DN} of the particles $\hat{M}_{k+1|k+1}^N$ over the value of the numerical state vector. At this step, particle weights are updated using a probability distribution function depending on a random noise that models uncertainty about continuous observations O_{k+1}^N.
- **(b)** A symbolic correction then computes a probability distribution Pr_{DS} over the symbolic states of the system, depending on discrete observations O_{k+1}^S and on Pr_{DN} making the process hybrid.

Finally, in order to update the complete predicted marking $\hat{M}_{k+1|k}$, a decision making method is required. The result of the whole state estimation process is the estimated marking at time $k+1$, $\hat{M}_{k+1|k+1} = \{\hat{M}_{k+1|k+1}^S, \hat{M}_{k+1|k+1}^N\}$.

Modified Particle Petri Nets have been originally designed to monitor hybrid system mission in (Zouaghi, Alexopoulos, Wagner, & Badreddin, 2011b). The main advantage provided by MPPN is the way they manage uncertainties. In this article, we will focus on a way to use them in a context of health monitoring.

2.4. Application to Health Monitoring

The main objective of the system health monitoring is to determine the health state of the system at any time (Chanthery & Ribot, 2013). Diagnosis is used to identify the probable causes of the failures by reasoning on system observation. Thus diagnosis reasoning consists in detecting and isolating faults that may cause a system failure. Results of the diagnosis function lead to the current health state of the system. To perform model-based health monitoring of a hybrid system, it is necessary to represent both behavioral model and degradation model of the system. We are interesting in representing changes in system dynamics when one or several anticipated faults happen. Thinking that way, we define a *health mode* by a discrete health state coupled to a continuous behavior. Then health state estimation partially relies on common techniques for continuous variable estimation. As long as the system does not encounter any fault, it is in a *nominal mode*. We assume that tracked faults are permanent. This means that once a fault happens, the system moves from a nominal mode to a *degraded mode* or faulty mode. Without repair, system evolution is unidirectional and ends with a *failure mode* whereas the system is not operational anymore. This evolution is illustrated in Figure 4.

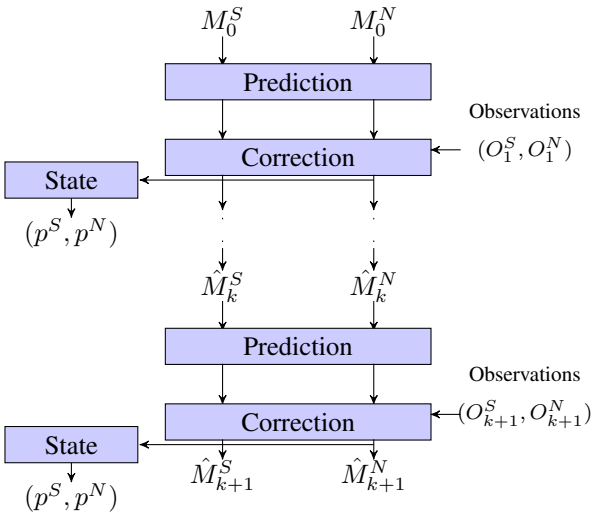


Figure 3. Hybrid sate estimation process of MPPN.

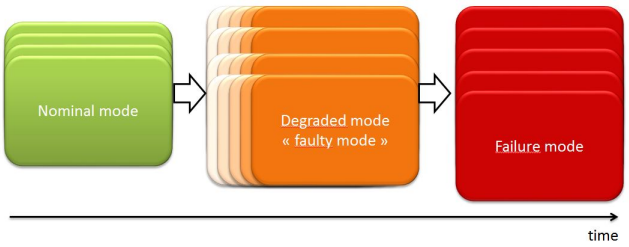


Figure 4. Unidirectional system evolution without maintenance or repair action.

Regarding the degradation model, we consider that faults in the system age depending on a stress level that is relative not to a behavior but to a health mode.

With the definition of the MPPN abstraction provided in previous sections, it is possible to model hybrid system behavior. Indeed, MPPN numerical places can be used to represent system dynamics, and symbolic places can be used to represent the different discrete health states of the system. Systems dynamics are then represented by differential equations. Thus, a hybrid state (p_i^S, p_j^N) will represent a health mode of the system. We designate by $Q = \{q_m\}$ the set of health modes of our system:

$$q_m = (p_i^S, p_j^N) \in Q \;\; if \;\; \exists t_l \in T, (p_i^S, p_j^N) \in (Post(t_l))^2 \tag{2}$$

where $Post(t_l)$ is the set of output places of t_l.

Using places that way, it becomes possible to use the symbolic conditions to model the occurrence of observable discrete events belonging to Σ_o and unobservable discrete events belonging to Σ_{uo} (faults, mission events, interaction with the environment, etc ...). $\Sigma = \Sigma_o \cup \Sigma_{uo}$ is defined as the set of discrete events of the system.

An example of a system behavioral model is described in Figure 5. In this example, the system has three different dynamics represented by p_5^N, p_6^N, p_7^N and four different health states p_1^S, p_2^S, p_3^S and p_4^S. By using Equation 2, five health modes are distinguishable. Health modes $q_1 = (p_1^S, p_5^N)$ and $q_2 = (p_1^S, p_6^N)$ are two nominal modes changing from the one to the other when condition $\Omega^S(t_1^S) = occ(e_1)$ or condition $\Omega^S(t_2^S) = occ(e_2)$ is satisfied. These conditions represent respectively the occurrence of observable events $e_1 \in \Sigma_o$ and $e_2 \in \Sigma_o$ supporting a change of behavior between p_5^N and p_6^N. Health modes $q_3 = (p_2^S, p_6^N)$ and $q_4 = (p_3^S, p_6^N)$ are two degraded modes reachable from health mode q_1 by satisfying the conditions $\Omega^S(t_3^S) = occ(f_1)$ and $\Omega^S(t_4^S) = occ(f_2)$ respectively. These two conditions represent respectively the occurrence of two unobservable fault events $f_1 \in \Sigma_{uo}$ and $f_2 \in \Sigma_{uo}$. Finally, $q_5 = (p_4^S, p_7^N)$ is a failure mode in which both f_1 and f_2 occurred and is reachable from the two degraded modes. Therefore $\Omega^N(t_5^S) = occ(f_1)$ is associated to the occurrence of f_1 and $\Omega^S(t_6^S) = occ(f_2)$ is associated with the occurrence of f_2.

While the design of the degradation model and its interaction with the behavioral model will be presented in Section 4, Section 3 will present a methodology to build a state tracker object called a diagnoser from the behavioral system model.

3. BEHAVIORAL DIAGNOSIS

In health monitoring, diagnosis is used to track system current health state. To do so, a common way is to generate a diagnoser of the system from the system model (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). The diagnoser is basically a monitor that is able to process any possible observable event on the system. It consists in recording these observations and providing the set of possible faults whose occurrence is consistent with these observations.

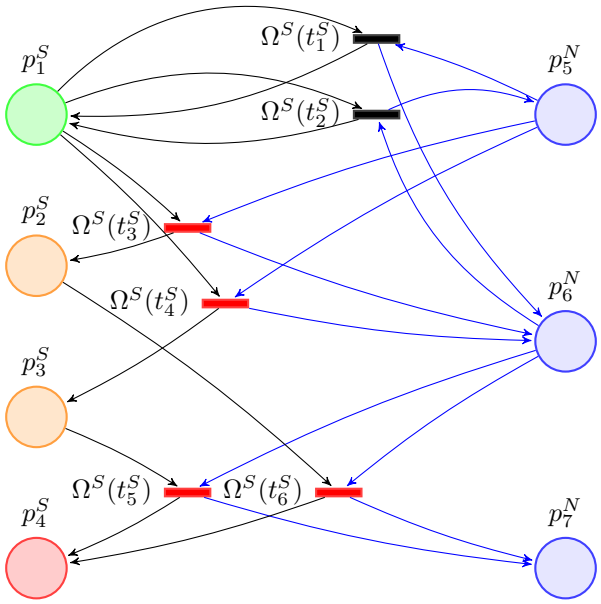


Figure 5. Example of system behavioral model using MPPN.

Concerning hybrid systems, one approach is to build a hybrid diagnoser (Bayoudh et al., 2008) from a hybrid automaton describing the system. The major idea is to abstract the continuous part of the system to only work with a discrete view of the system. This abstraction is done by using consistency tests, that take the form of a set of analytical redundancy relations (ARR). The diagnoser method is then directly applied on the resulting discrete event system. In previous works (Chanthery & Ribot, 2013), we extended this approach in order to integrate diagnosis and prognosis for hybrid systems. The main drawback of this approach is that the DES oriented diagnosis framework seems not the best suited for the incorporation of the highly probabilistic prognosis task. With the MPPN representation, we succeed in capturing all the uncertainties about the state knowledge, but also about the observations. Consequently, we have to develop a new diagnoser build from an MPPN. Moreover, the classical diagnoser is a finite state machine. If this theoretical object is very interesting for studying properties on system, like diagnosability or controllability, it is absolutely not suited for embedded systems, because the number of states of the diagnoser explodes for large models. Consequently, we choose to build a diagnoser based on a MPPN model for the following reasons:

- there is no lack of information during the diagnoser generation,
- MPPN model captures all the uncertainties,
- this representation is more compact than hybrid automa-

ton description, so the problem of embeddability of the diagnoser is reduced.

The diagnoser takes as input the MPPN specifying the behavior of the system and the set of online observations on the system. The output of the diagnoser is an estimation of the health state of the system. Next sections describe how to generate a diagnoser from an MPPN specifying the behavior of a system, then define what is finally called a diagnosis and how this object may be used for health monitoring.

3.1. Diagnoser Generation Based on MPPN

The goal of this section is to generate a MPPN that is able to monitor the system current health state thanks to the observations. Let suppose that the MPPN specifying the behavior of the system is a tuple $< P, T, Pre, Post, X, F, \gamma, \Omega, M_0 >$ as defined in Section 2.1. The set of places of the diagnoser remains the same as the one of the system. Concerning the transitions, there are two aspects to take into account.

First, it is necessary to follow the continuous behavior of the system with information issued from the observed variables of the system. A set of analytical redundancy relations (ARR) can be generated from the set of differential equations C of the system model. In the linear case, ARRs can be computed by using the parity space approach (Staroswiecki & Comtet-Varga, 2001). The parity space approach has been extended to multi-mode systems in (Cocquempot, El Mezyani, & Staroswiecki, 2004). In our case, a relation ARR_i is associated to each numerical place p_i^N. A numerical condition $\Omega^N(t_l)$ associated with a transition t_l linking two numerical places p_i^N and p_j^N carries ARR_{ij} satisfaction test, with $(i,j) \in \{1,...,|P^N|\}^2$ and $l \in \{1,...,|T|\}$. This means that $\Omega^N(t_l)(\pi)$ is satisfied when ARR_{ij} is satisfied for π. ARRs are satisfied if the observations satisfy the model constraints. Since ARRs are constraints that only contain observable variables, they can be evaluated online with the incoming observations given by the sensors. It is thus possible to check the consistency of the observed system behavior with the predicted one.

Secondly, because the diagnoser only captures the observable behavior of the system, a condition representing the occurence of an unobservable discrete event would never be satisfied. Consequently, all the symbolic conditions representing the occurences of unobservable events are removed from Ω without loss of information. Concerning the observable discrete part of the system, occurrences of observable discrete events will be used as symbolic condition triggers.

Once the system behavioral model is defined and all numerical conditions are computed from the ARRs generation, the corresponding diagnoser can be generated with the following steps:

Step 1: Add corresponding numerical conditions $\Omega^N(t_j^S)$ to every symbolic transition $t_j^S \in T^S$, with $j \in \{1,...,|T|\}$. As a result, the symbolic transition t_j^S will be upgraded into a mixed transition $t_j^M \in T^M$.

Step 2: Remove, from any mixed transition $t_j^M \in T^M$, symbolic conditions $\Omega^S(t_j^M)$ covering the occurrence of an unobservable event, because these conditions would never be satisfied. Consequently, the mixed transition t_j^M is transformed in a numerical transitions $t_j^N \in T^N$.

Ambiguity: Hybrid system diagnosis consists in determining the health state of the system wherein observations are consistent. Diagnosis challenge is the ability to diagnose anticipated but unobservable faults in the system. In this context, modeling unobservable events can lead to ambiguity in the diagnoser. Indeed, the occurrence of several faults that can not be distinguishable with the observations of the systems will lead to ambiguous health states for the diagnoser. Therefore, a third step is needed during the diagnoser generation to track ambiguity. To do so, it is necessary to define a merger property to merge two numerical transitions. Two numerical transitions are mergeable if they are conditioned by the same dynamics change and if they share the same symbolic places in their sets of inputs places. In a more formal way, let $Pre(t_j)$ be the set of input places of a transition $t_j \in T$:

$$Pre(t_j) = \{p_i | Pre(i,j) \neq 0, i \in \{1,...,|P|\}\} \quad (3)$$

As well, $Post(t_j)$ is the set of its output places:

$$Post(t_j) = \{p_i | Post(i,j) \neq 0, i \in \{1,...,|P|\}\} \quad (4)$$

Definition 1 *Two numerical transitions $(t_i^N, t_j^N) \in (T^N)^2$, with $(i,j) \in \{1,...,|T^N|\}^2$ and $i \neq j$ are mergeable if :*

$$(Pre(t_i^N) = Pre(t_j^N)) \ \wedge \ (Post(t_i^N) \cap P^N \cap Post(t_j^N) \neq \emptyset) \quad (5)$$

Note that condition (5) implies that the two transitions share the same numerical condition: $\Omega^N(t_i^N) = \Omega^N(t_j^N)$.

Step 3: Merge all mergeable transitions while there is at least two mergeable transitions using the following merging definition:

Definition 2 *The merging of two mergeable numerical transitions $(t_i^N, t_j^N) \in (T^N)^2$, with $(i,j) \in \{1,...,|T^N|\}^2$ and $i \neq j$ is defined by two steps as follows:*

(1) *Creation of a new transition t_{ij}^N characterized by:*

$$\begin{cases} Pre(t_{ij}^N) & = & Pre(t_i^N) \\ Post(t_{ij}^N) & = & Post(t_i^N) \cup Post(t_j^N) \\ \Omega^N(t_{ij}^N) & = & \Omega^N(t_i^N) \end{cases} \quad (6)$$

(2) *Introduction of t_{ij}^N and deletion of t_i^N and t_j^N in T:*

$$T = (T \backslash \{t_i^N, t_j^N\}) \cup \{t_{ij}^N\} \quad (7)$$

The resulting diagnoser of the model in Figure 5, after computing the third steps above, is presented in Figure 6.

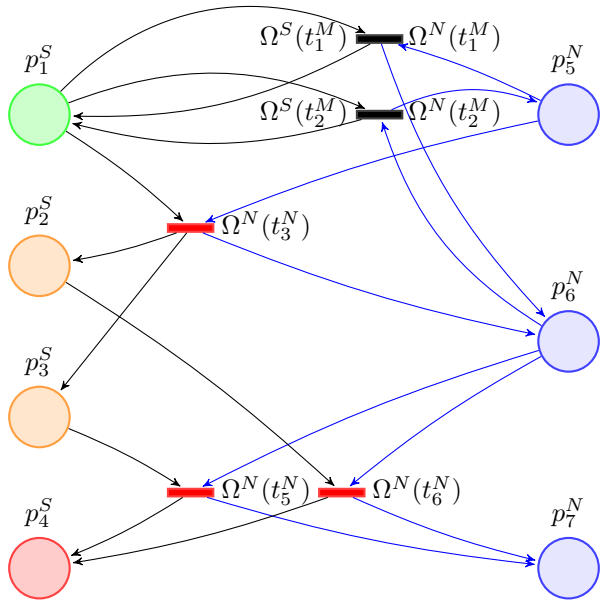


Figure 6. Example of diagnoser of system using MPPN.

In Figure 6, performing Step 1 has generated numerical condition Ω^N to every transition. Indeed, all transitions where supported by a change of dynamics that can be observed with the generation of the ARR. After this step on this example, all transitions are upgraded into mixed transitions. As there were unobservable events, symbolic conditions associated with the occurrence of f_1 and f_2 have been removed from the diagnoser model during Step 2, transforming t_3, t_4, t_5 and t_6 into numerical transitions. Finally, because transitions t_3 and t_4 were generating a change of dynamics from p_1^N to p_2^N, they were mergeable and thus have been merged into one single numerical transition t_3^N.

3.2. Behavioral Diagnosis Results

The behavioral diagnosis is defined at each clock tick as the state of the diagnoser. By using the MPPN, the diagnosis Δ_k at time k is the distribution of health mode believes that depends on particle values and weights and is deduced from the marking of the diagnoser at time k :

$$\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N\} \quad (8)$$

The marking $\hat{M}_k$ indicates the belief on the fault occurrences. It gives the same information than a classical diagnoser mode in terms of faults occurrences, with the same ambiguity. The difference is that in a classical diagnoser, every possible diagnosis has the same belief degree. With MPPN-based diagnoser, the ambiguity is valued by the knowledge about the weights of each particle of the marking.

Consequently, using the diagnosis results for health management becomes easier. Indeed, in the case of classical diagnoser, it is very difficult to "choose" a belief state for the system in case of decision making. It is then very important to obtain the less ambiguous diagnosis as possible. In the case of MPPN-based diagnoser, each possible state of the system is valued, so it is easy to evaluate the more probable state at each clock tick.

4. DEGRADATION DIAGNOSIS

The previous part describes a way to use MPPN to monitor health state of the system based on its behavioral model. It is often interesting to take into account another level of representation to illustrate a different level of dynamics, or a more aggregate view of the system. For instance, in the framework of health monitoring, it is worth to look at the system at another level to take into account the degradation dynamics. Getting some information about the degradation of the system is a huge advantage for elaborating a more precise diagnosis and to perform prognosis.

Next sections describes what we call *Hybrid Particle Petri Nets* (HPPN). HPPN give a theoretical framework to represent MPPN at a higher level called the *hybrid level*. The purpose of this hybrid level is to represent some hybrid states characteristics, and not only continuous behavior or discrete state. A set of dynamics equations is used to follow hybrid information we are focused on. To point out this new hybrid level, we assume that places, transitions, conditions and tokens used in Section 2 and Section 3 are part of the *behavioral level*. Because of the new hybrid level, the enriched formalism is called Hybrid Particle Petri nets. The set of dynamics equations we focus on with the hybrid level represent component degradation laws, that depend on the health modes of the system. The update of the degradation value at each clock tick defines a degradation diagnosis function. The application of HPPN for health monitoring is then illustrated on an example.

4.1. Hybrid Level

A Hybrid Particle Petri Net is described as an enriched MPPN $< P, T, Pre, Post, X, C, H, \mathcal{F}, \gamma, \Omega, M_0 >$ where:

- P is the set of places, partitioned into numerical places P^N, symbolic places P^S and hybrid places P^H.
- T is the set of transitions (numerical T^N, symbolic T^S, mixed T^M and hybrid T^H).
- $H \subset \Re^n$ is the state space of the hybrid state vector.
- $\mathcal{F}$ is the set of dynamics equations of the system associated with hybrid places, representing hybrid state evolution.
- Ω is the set of conditions associated with the transitions (numerical Ω^N and symbolic Ω^S and hybrid Ω^H).

Hybrid places are used to compose the hybrid level and represent possible hybrid states of system. In HPPN, a hybrid state is a couple (p_i^S, p_j^N). For the sake of clarity in the paper, we will use $p_l^H = (p_i^S, p_j^N)$ to indicate that hybrid place p_l^H represents the hybrid state (p_i^S, p_j^N). Because hybrid states are combinations of symbolic places and numerical places, the set of hybrid states for a given behavioral model is always finite. However, only couples that are part of the set of output places of the same transition are considered as hybrid states. Formally:

$$p_l^H = (p_i^S, p_j^N) \in P^H \; if \; \exists t_m \in T, (p_i^S, p_j^N) \in (Post(t_m))^2 \tag{9}$$

Hybrid states that do not satisfy Condition 9 are considered as *intermediate states*. This means there is no information in the model about these hybrid states.

A hybrid place is marked by hybrid tokens $h_k^i = [s_k^i, \eta_k^i]$ with $i \in \{1, ..., |M_k^H|\}$ where M_k^H is the set of all the hybrid tokens in the net at time k. A hybrid token is defined by a couple $s_k^i = (\delta_k^j, \pi_k^l)$ of tokens running in the behavioral level and its corresponding hybrid state vector $\eta_k^i \in H$. The whole marking at time k of the HPPN is $M_k = \{M_k^S, M_k^N, M_k^H\}$. Now that hybrid tokens have been described, we are going to detail their creation and deletion rules.

Creation: Because of their dependencies on configurations and particles, new hybrid tokens are created at the same time of creation of a configuration or a particle. If a hybrid token h^i depends on a particle π^l that is duplicated during the particle filter step in a new particle π'^l, then h^i is also duplicated in h'^i but h'^i depends on the new particles π'^l.

Deletion: A hybrid token h^i depending on a configuration δ^j and a particle π^l is deleted when δ^j or π^l is deleted during the online process of the behavioral level.

Considering the two rules above, the hybrid level online process totally depends on the behavioral level online progress. However, the two processes are simultaneous.

Any hybrid place is linked with all other hybrid places through a hybrid transition $t_j^H \in T^H$.

$\forall p_i \in P$, $M_k(p_i)$ is the set of tokens in p_i at time k and $m_k(p_i) = |M_k(p_i)|$ is the number of tokens in p_i at time k.

Definition 3 *A hybrid transition $t_j^H \in T^H$ is fire-enabled at time k if:*

$$\exists p_i^H \in Pre(t_j^H), \quad m_k(p_i^H) \geq Pre(i,j) \tag{10}$$

A hybrid place is associated with a set of dynamics equations representing a hybrid state characteristic. The idea is to let evolve a hybrid token $h^i = [s^i, \eta^i]$ in the hybrid level in accordance to the symbolic and numerical places in which are evolving its associated configuration δ_k^j and its associated particle π_k^l, with $s^i = (\delta^j, \pi^l)$.

To formally define the firing of hybrid transitions, we need to define the following notations. $P(\delta^j) = p_j^S$ and $P(\pi^l) = p_l^N$ denote the projections of δ^j and π^l on the set of places P. Then, $P(s^i) = (p_j^S, p_l^N)$ denote the hybrid place of a couple $s^i = (\delta^j, \pi^l)$.

Every hybrid transition carries a hybrid condition $\Omega^H(t_j^H)$ which is satisfied if $\Omega^H(t_j^H)(h^i) = 1$. Hybrid tokens h^i are moved to another hybrid place p'^H if $P(s^i) = p'^H$. Formally:

$$\forall h^i = [s^i, \eta^i], \quad \Omega^H(t_j^H)(h^i) = \begin{cases} 1 & if \;\; P(s^i) = p'^H \\ 0 & otherwise \end{cases} \tag{11}$$

$\mathcal{S}_k^H(p^H)$ is the set of hybrid tokens in p^H satisfying the condition $\Omega^H(t_j^H)$ at time k:

Equation 11 implies that every transition t_j^H has only one hybrid output place p'^H and sees all the other hybrid places p^H as input places. More formally:

Definition 4 *The firing of a fire-enabled hybrid transition $t_j^H \in T^H$ at time k is defined by:*

$$\begin{cases} M_{k+1}^H(p^H) = M_k^H(p^H) \backslash \mathcal{S}_k^H(p^H) \\ M_{k+1}^H(p'^H) = M_k^H(p'^H) \cup \mathcal{S}_k^H(p^H) \end{cases} \tag{12}$$

An example of hybrid transition firing in a hybrid level is shown in Figure 7. In the example, there are two hybrid places $p_1^H = (p_1^S, p_1^N)$ and $p_2^H = (p_2^S, p_2^N)$. At time k, the two hybrid tokens $h_k^1 = [s_k^1, \eta_k^1]$ and $h_k^2 = [s_k^2, \eta_k^2]$ are following the characteristic of the hybrid state represented by p_1^H, so hybrid transitions t_1^H and t_2^H are fire-enabled. $P(s_k^1)$ is (p_1^S, p_1^N) but $P(s_k^2)$ is (p_2^S, p_2^N) so $\Omega^H(t_2^H)(h^2)$ is satisfied and h^2 is moved through t_2^H. Thus, h^2 is in the hybrid place p_2^H at time $k+1$ and follows the characteristic of the hybrid state (p_2^S, p_2^N).

This enrichment evolves all the possible hybrid states of the system alongside according to their corresponding laws. Indeed, because tokens in the behavioral level are changing of places during the prediction step (see Section 2.3 (1)), hybrid

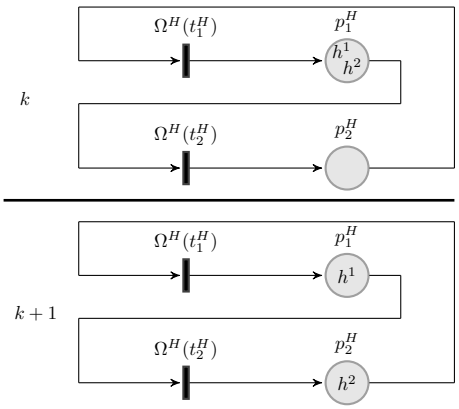


Figure 7. Illustration of firing rules of hybrid fire-enabled transitions.

tokens are simultaneous changing of places and their values are updated as follows:

$$\forall h^i_{k+1|k} \in \hat{M}^H_{k+1|k}(p^H_j), \quad \eta^i_{k+1} = F^j_{k+1}(\eta^i_k) \tag{13}$$

where $F^j_k \in \mathcal{F}$ is the set of dynamics equations associated with the hybrid place p^H_j. Because η^i_{k+1} depends on η^i_k, the continuity of the value η^i can be ensured. Figure 8 illustrates the evolution of the value η^2 of hybrid token h^2 of Figure 7. It shows that η^2_{k+1} is computed with the dynamics equation F^2_{k+1}. F^2_{k+1} is associated with p^H_2 and depends on η^2_k the value of η^2 at time k. This dependency ensures the continuity between F^1_k and F^2_{k+1} at time $k+1$.

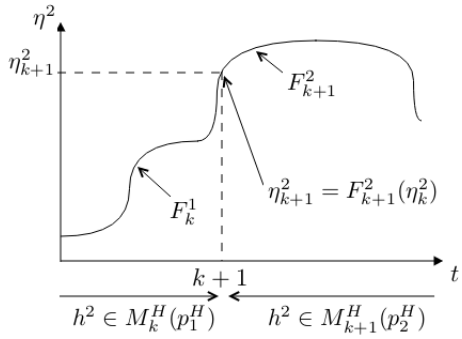


Figure 8. Illustration of the continuities of hybrid token values.

If $\mathcal{F}$ is not empty, the values η^i_k can be taken into account in the decision making process at time k that determine the marking at time $k+1$ of the behavioral level.

If the set of hybrid characteristics $\mathcal{F}$ is empty, the hybrid level directly monitors the hybrid state of the system over a distribution of hybrid tokens considering the particles weights. Moreover, considering a HPPN A, if $\mathcal{F} = \emptyset$ hybrid tokens has no value ($\eta^i = 0$) so they can be considered as configurations in another HPPN B. As well, if $\mathcal{F} \neq \emptyset$, values η^i of hybrid tokens evolve depending on the hybrid places and thus they can be considered as particle for HPPN B. By this way, hybrid tokens can go through a particle filter, making the hybrid level values having an effect on the configurations and particles of the behavioral level of HPPN A. Following this reasoning, we understand that the HPPN formalism is recursive. MPPN/HPPN can model hybrid systems, so by using only numerical places, numerical transitions and particles, its is possible to monitor continuous systems. As well, by using only symbolic places, symbolic transitions and configurations, it is possible to monitor discrete systems. This means that the HPPN formalism is also generic and can model different kind of systems such as heterogeneous systems. Finally, because HPPN is recursive, generic and can model discrete, continuous and hybrid systems, HPPN can be considered as a holistic method.

4.2. HPPN for Health Monitoring

This section introduces a way to represent uncertainty about degradation for each health mode of the system using probability measures.

The system description is enriched with a set of degradation laws modeling the degradation depending on hybrid state stress levels. The set of degradation laws is supposed to be accurately known. $\mathcal{F} = \{F^{q_m}, q_m \in Q\}$ is the set of degradation laws associated with health modes of the system. F^{q_m} is a vector of degradation laws for each anticipated fault in the health mode $q_m = (p^S_i, p^N_j)$. For example, in a system where n_f faults are considered:

$$F^{q_m}(t) = \begin{bmatrix} f_1^{q_m}(t) \\ f_2^{q_m}(t) \\ \vdots \\ f_{n_f}^{q_m}(t) \end{bmatrix} \tag{14}$$

where $f_j^{q_m}$ represents the probability distribution of the fault f_j at any time in the health mode q_m.

In the context of health monitoring, we need the formalism of the hybrid level to include health mode degradation laws in our model. We propose to consider health modes as hybrid states of an HPPN. Thus health modes are represented by hybrid places (see Section 2.4) and the set of degradation laws will be the set of dynamics equations associated with hybrid places.

Figure 9(b) represents the degradation laws model of the example of Figure 5. This system has five health modes (see Section 2.4), thus the corresponding hybrid level has five hybrid places $p^H_8 = (p^S_1, p^N_5)$, $p^H_9 = (p^S_1, p^N_6)$, $p^H_{10} = (p^S_2, p^N_6)$, $p^H_{11} = (p^S_3, p^N_6)$ and $p^H_{12} = (p^S_4, p^N_7)$. Therefore five hybrid transitions t^H_7, t^H_8, t^H_9, t^H_{10} and t^H_{11} deliver accesses to the five

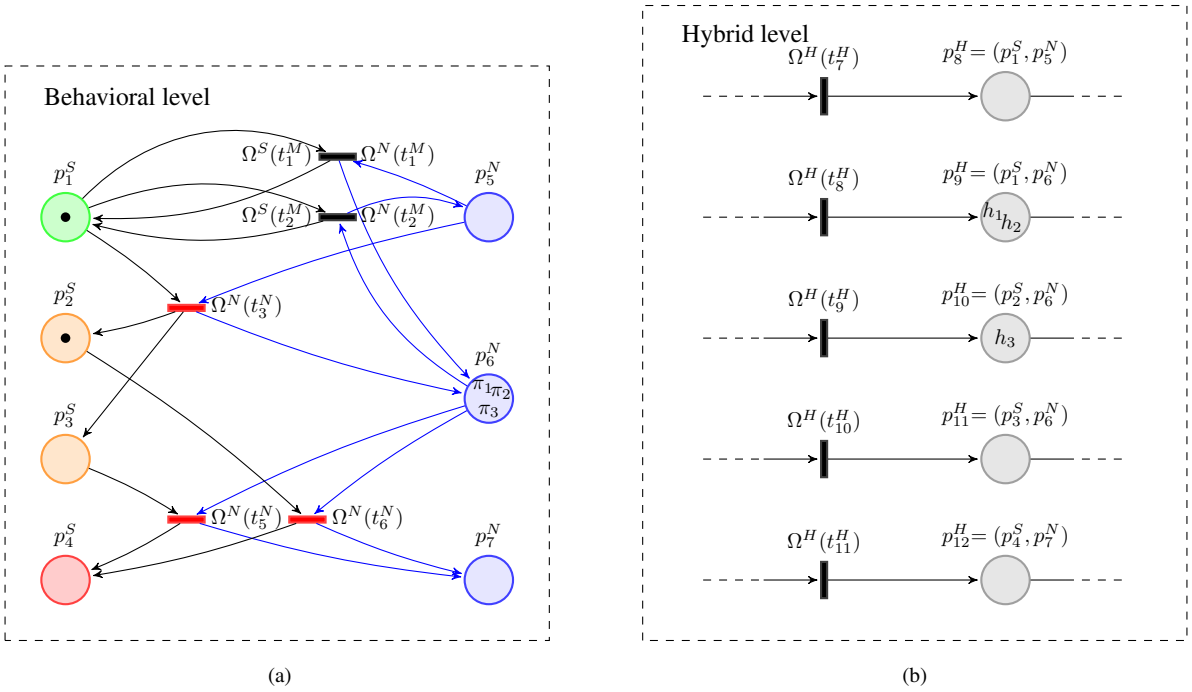


Figure 9. Example of diagnoser of system using HPPN.

hybrid places when associated hybrid conditions are satisfied (Equation 11). All the transitions are not represented in the figure because of the complexity of the representation.

4.2.1. Diagnoser Generation Based on HPPN

The diagnoser generation step does not change the degradation model during its computation. The degradation model is added to the behavioral diagnoser (Section 3.1) as a hybrid level. The result of the whole generation step is a HPPN-based diagnoser that monitors both the behavior and the degradation of the system.

Figure 9 shows the complete diagnoser of the system example presented in this paper. It illustrates the interactions between the behavioral level (a) and the hybrid level (b) of the diagnoser. Two configurations and three particles are running in the behavioral level. One configuration is in the symbolic places p_1^S and the other one in the symbolic place p_2^S. All three particles π^1, π^2 and π^3 are in numerical place p_6^N. Therefore, three hybrid tokens are running in the hybrid level. h^1 and h^2 are in the hybrid place p_9^H because they are linked to configuration in p_1^S and respectively π^1 and π^2. However, h^3 is in the hybrid place p_{10}^H because it is linked to the configuration in p_2^S and π^3.

4.2.2. Diagnosis Results

Using HPPN-based diagnoser, the diagnosis Δ_k of the system at time k is the complete marking of the diagnoser, indicating the distribution of health mode believes depending on particle values and weights and hybrid token values:

$$\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N, \hat{M}_k^H\} \tag{15}$$

The marking $\{\hat{M}_k^S, \hat{M}_k^N\}$ represents the belief on the health modes through a probabilty distribution. The marking $\hat{M}_k^H$ represents a degradation distribution over the health modes. Because each hybrid token depends on a particle and a configuration, its degradation value is linked with the belief of its health mode. Consequently, the belief and the degradation value can be correlated in case of decision making in the context of health management.

5. CONCLUSION AND FUTURE WORK

This paper formally introduces the HPPN approach to model the monitoring of hybrid systems. The MPPN method is enriched to consider another level to represent a hybrid dynamics. The method takes into account uncertainty about the knowledge of the system and uncertainty during the online process, such as continuous and discrete observations. The article then proposes to use HPPN to build a diagnosis methodology in a health monitoring context. HPPN can be used to model a diagnoser to monitor both discrete and continuous behaviors of the system, but also to consider the system degradation depending on the hybrid state of the system. The methodology is illustrated with an academic example. The building of such a diagnoser is a first step to perform

prognosis and health management of hybrid systems under uncertainty. Moreover, diagnosis results can be used as probability distributions for decision making.

In future works, we will implement this work and test it on an embedded system. The prognosis methodology will be formally described considering the InterDP framework introduced in (Chanthery & Ribot, 2013) that interleaves diagnosis and prognosis methods to let results be more accurate.

REFERENCES

Bayoudh, M., Travé-Massuyes, L., & Olive, X. (2008). Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the IFAC World Congress* (pp. 7265–7270). Seoul, Korea.

Chanthery, E., & Ribot, P. (2013). An integrated framework for diagnosis and prognosis of hybrid systems. In 3^{rd} *Workshop on Hybrid Autonomous System (HAS).* Roma, Italy.

Cocquempot, V., El Mezyani, T., & Staroswiecki, M. (2004). Fault detection and isolation for hybrid systems using structured parity residuals. In 5^{th} *Asian Control Conference.* (Vol. 2, pp. 1204–1212).

Gaudel, Q., Chanthery, E., Ribot, P., & Le Corronc, E. (2014). Hybrid systems diagnosis using modified particle petri nets. In *Proceedings of the* 25^{th} *International Workshop on Principles of Diagnosis (DX'14).* Graz, Austria.

Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the* 11^{th} *Annual IEEE Symposium on Logic in Computer Science* (pp. 278–292).

Lesire, C., & Tessier, C. (2005). Particle petri nets for aircraft procedure monitoring under uncertainty. In *Applications and Theory of Petri Nets* (pp. 329–348). Springer.

Ribot, P., Pencolé, Y., & Combacau, M. (2013). Generic characterization of diagnosis and prognosis for complex heterogeneous systems. *International Journal of Prognostics and Health Management*, *4*.

Roychoudhury, I., & Daigle, M. (2011). An integrated model-based diagnostic and prognostic framework. In *Proceedings of the* 22^{nd} *International Workshop on Principle of Diagnosis (DX'11).* Murnau, Germany.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555–1575.

Staroswiecki, M., & Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, *37*(5), 687–699.

Vinson, G., Ribot, P., Prado, T., & Combacau, M. (2013). A generic diagnosis and prognosis framework: application to permanent magnets synchronous machines. In *IEEE Prognostics and System Health Management Conference (PHM)* (pp. 1039–1044). Milano, Italy.

Zouaghi, L., Alexopoulos, A., Wagner, A., & Badreddin, E. (2011a). Modified particle petri nets for hybrid dynamical systems monitoring under environmental uncertainties. In *IEEE/SICE International Symposium on System Integration (SII)* (pp. 497–502).

Zouaghi, L., Alexopoulos, A., Wagner, A., & Badreddin, E. (2011b). Probabilistic online-generated monitoring models for mobile robot navigation using modified petri net. In 15^{th} *International Conference on Ad-*

Towards Performance Prognostics of a Launch Valve

Glenn Shevach[1], Mark Blair[2], James Hing[3], Larry Venetsky[4], Everard Martin[5], John Wheelock[6]

[1,2,3,4]*Naval Air Warfare Center-Aircraft Division Code 4.8.1.4, Lakehurst, NJ, 08733, USA*

glenn.shevach@navy.mil
mark.r.blair@navy.mil
james.hing@navy.mil
larry.venetsky@navy.mil

[5,6]*Naval Air Warfare Center-Aircraft Division Code 4.8.2.1, Lakehurst, NJ, 08733, USA*

everard.martin@navy.mil
john.wheelock@navy.mil

ABSTRACT

Due to its criticality in aircraft carrier steam catapult operations, the performance of the Launch Valve is monitored using timer components to determine the elapsed time for the valve to achieve a set opening distance. Significant degradation in performance can lead to loss in end speed of the catapult and result in loss of aircraft / lives. This paper presents a method of using existing timing data for anomaly detection and predicting when maintenance is required (MIR) for a Launch Valve. Features such as mean and standard deviation of timing values are extracted from clock time data to detect anomalies. Neyman-Pearson Criterion and Sequential Probability Ratio Testing are used to formulate a decision on the degraded state. Once an anomaly is detected, an observation window of the previous *N* filtered samples are used in a risk sensitive particle filter framework. The resulting distribution is used in the prediction of shots until MIR. Performance degradation is extracted from training data and modeled as a third order polynomial. The algorithm was tested on two test sets and validated by Subject Matter Experts (SMEs) supplying the data. An Alpha-Lambda performance metric shows the time predictions until MIR fall inside an acceptable performance cone of 20% error.

1. INTRODUCTION

Steam catapults are among the oldest and most maintenance-intensive systems in the Navy. The steam catapult is a system that launches aircraft from an aircraft carrier by releasing built up steam pressure behind a shuttle that pulls the aircraft along the deck. This critical system is largely unchanged from the 1940's – steel, steam and hydraulics that will be with us for the next 40 years. Yet catapults need to perform flawlessly and maintain a system reliability of 99.9999 or the result is loss of aircraft and lives. (Reliability of 99.9 = 140 lost aircraft per year; 99.99 = 14 lost aircraft per year) The Fleet ensures these systems are reliable, but at a very high cost in terms of spares, overhauls and manpower. A reduction in costs could be achieved through prognostic and health management (PHM) methods. The ability to predict impending failures or needed maintenance of these systems in real time, could reduce total ownership costs by decreasing maintenance, inventory, and down time.

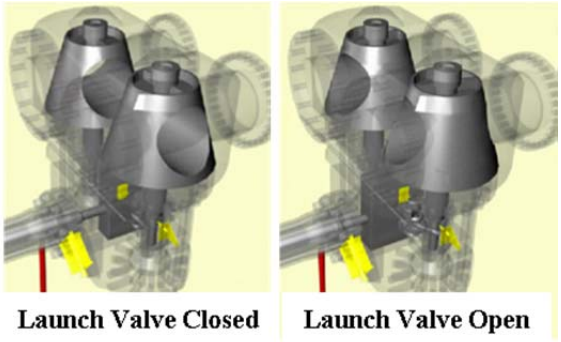


Figure 1: View of Launch Valve in closed and open positions

The Low Loss Launch Valve (LLLV), hereby known as the Launch Valve, is a hydraulically controlled valve and provides a means for controlling the steam pressure in the catapult power cylinders for launching aircraft (shown in Figure 1). In order to launch the full range of fleet aircraft, the energy of each launch must be tailored for the specific aircraft type and weight, as well as the current wind over deck (WOD) conditions. This is accomplished by adjusting

Glenn Shevach et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. NAVAIR Public Release 2014-400. Distribution Statement A – "Approved for public release; distribution is unlimited"

the opening rate of the Launch Valve to introduce the proper amount of steam. Because of its high reliability requirement, the Launch Valve is designed to have one of the highest operational availabilities compared to all other components within the catapult sub-system. Degradation not being identified quickly can result in additional degradation which could cause a significant loss in end speed and an urgent halt to operations until the degradation was corrected. Insufficient catapult end speed can result in loss of aircraft / lives.

The fleet checks the Launch Valve performance during launching operations with pre-op Blow-Through-No-Loads (BTNL) (no aircraft connected to the catapult shuttle). These times are manually read by an operator, transcribed in a paper log, and typed into electronic spreadsheets hours later. The process is prone to inscription errors. A detailed analysis of Launch Valve performance is manually reviewed upon submission at the conclusion of each month. Subject Matter Experts (SMEs) review clock times to sift out inscription errors and advise for further maintenance actions. This time consuming process relies heavily on the historical knowledge and judgment to decide when a Launch Valve is starting to show signs of degradation. The delay in detailed analysis leaves the potential for degradation to go unnoticed and uncorrected. Continuous real time monitoring of the Launch Valve performance could detect trends in degradation before they reach a critical point.

This paper presents efforts towards the ultimate goal of giving the fleet real time prognostics and health monitoring of the Launch Valve performance during aircraft operations. The algorithm utilizes available Launch Valve clock timing data to detect anomalies and predict when maintenance is required (MIR). Probabilistic techniques are used to detect, with minimum false alarms, the degradation in performance of a Launch Valve and prognostic techniques are used to predict when the degradation will cross a "maintenance needed" threshold. A unique quality to this data is that it is comprised of manually entered time. An operator reads the output of the timers and manually inputs it into a spreadsheet. The algorithm presented takes in timing data over a series of Launch Valve openings that are susceptible to user inscription error.

The paper is structured as follows: Section 2 discusses related works on prognostics and health monitoring of valves. Section 3 provides background information of Launch Valve operation. Section 4 provides the theoretical background for feature extraction, anomaly detection, degradation modeling, and forecasting techniques. Section 5 presents results and discussion using real world Launch Valve timing data and Section 6 concludes the paper with a summary of the findings and future work.

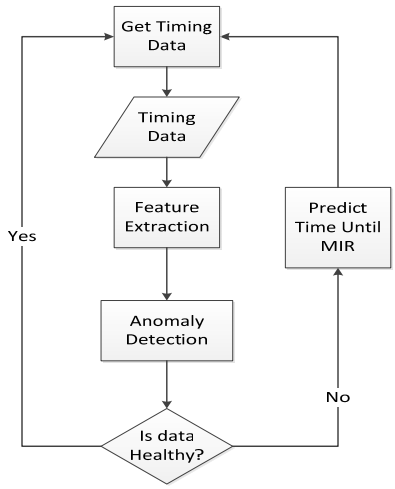


Figure 2: Flow chart for Launch Valve Prognostics

2. RELATED WORKS

Two notable works are related to this paper's efforts. Gomes et. al. developed a health monitoring system for a pneumatic valve using a Probability Integral Transform based technique (Gomes 2010) and Daigle et. al. developed a model-based prognostics approach for pneumatic valves (Daigle 2011). While the Launch Valve in this work is hydraulically controlled, the methods used for pneumatic valve PHM are quite relevant. Diagle et. al. used a Probability Integral Transform to calculate an index of dissimilarity between pressure distributions of monitored and baseline (healthy) valve performance. They were able to use this index of dissimilarity feature to detect increasing degradation and failure of a valve. There was no prediction to failure presented. Timing data of the valve was not utilized. Daigle et. al. constructed a detailed physics-based model of a pneumatic valve that includes models of different damage mechanisms. They use time for the valve to open and close to perform the prognostics. In their work, they focused on the prediction portion of the work and started predictions at pre-defined known points in the historical data where degradation was observed.

3. LAUNCH VALVE OPERATION

The Launch Valve has two (2) clock switches, Clock No.1 and No.2 that are used to measure the time it takes the valve to open 23% and 60% of full open respectively. The beginning portion of launch valve stroke is very dynamic which leads to too much clock time variation in Clock No. 1 to be used as a performance indicator. Clock No. 2 provides less variation in clock times since it measures later in the valve stroke and is therefore used as a performance

indicator. Currently, the Launch Valve performance is monitored by the fleet using Launch Valve Clock No. 2 times from the two daily pre-operational Blow Through No Load (BTNL) launches. The times are compared to limits established in the applicable Maintenance Requirement Card. The fleet conducts both a shot by shot (real time) and long term trend evaluation of the BTNL clock times. NAVAIR Lakehurst also conducts a more detailed analysis of the Launch Valve performance using data (BTNL and aircraft) via the Automated Shot and Recovery Log (ASRL) provided by the fleet.

Degradation in performance of the Launch Valve can be assessed through analysis of this timing data. Performance degradation of the Launch Valve can be caused by increased friction due to loss of lubrication, other internal components providing high friction loads, or parameters outside the normal operating range. Slower clock times are representative of a valve experiencing high internal friction. Faster clock times are representative of a valve leakage in hydraulic fluid downstream. Other factors unrelated to performance are misalignment of the valve and body seat due to surface wear and degraded gasket condition. It can be difficult and costly to install sensors to monitor conditions such as lubrication, wear, gasket condition, etc. This is especially true in these cases where the Launch Valve already exists in a catapult system and cannot be modified. Therefore, a health management solution must be implemented using limited data and feature sets.

4. APPROACH

Figure 2 shows a flow chart for the process that the proposed prognostics algorithm follows.

4.1. Data Preparation

In its current state, the Launch Valve timing data requires some pre-processing by SMEs prior to being fed into the prognostics algorithm. Future work will look to automate the pre-scrubbing process. Raw Clock 2 data contains timing of all launches and blow through no loads. Launches with a low capacity selector valve (CSV) setting have to be identified and removed from the data because CSVs below a specific value do not tend to achieve the Clock 2 switch prior to the “launch complete” signal closing the Launch Valve. This results in inaccurate timing. After this scrub, clock times are compared to existing Clock 2 vs CSV curve baseline (4th order poly fit line) to determine "variation". A 4^{th} order polynomial was found to provide the best fit of the clock times for the range of CSV settings from aircraft operations based on historical data. The next step is the manual review of the data to identify if any shifts in the data occurred signifying a potential shift in the baseline is necessary. Over the life of the catapult the limit switches timing the opening of the Launch Valve will be replaced several times which can cause a shift in the data. If a shift

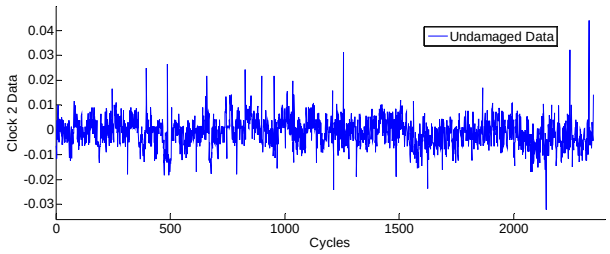


Figure 3: Good performance data of opening times of a Launch Valve Over a One Year Period.

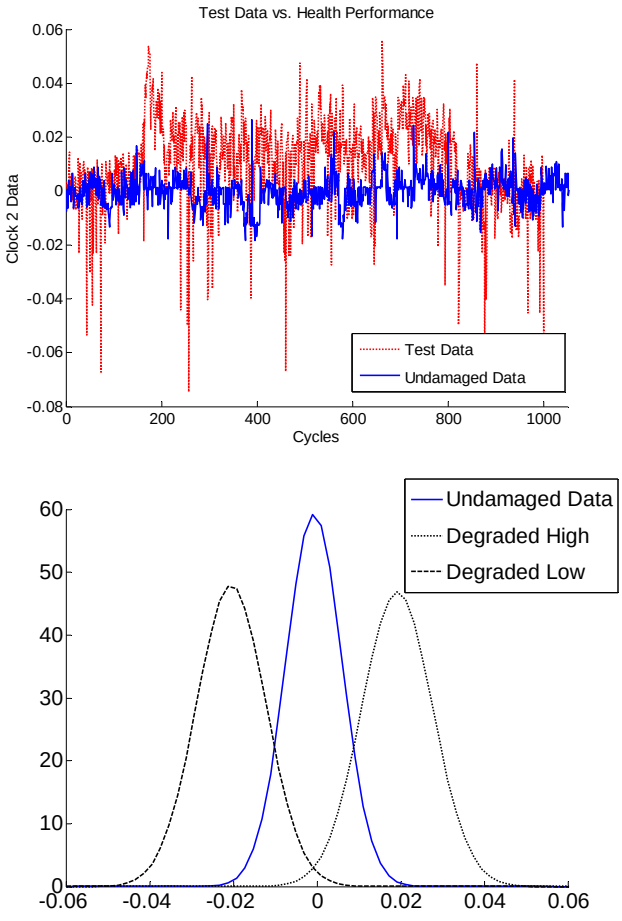


Figure 4: Top) Healthy Data (blue solid) vs. Degraded Data (red dashed), Bottom) Gaussian distributions of good performance data (blue solid) and degraded performance high / low. Degraded High means longer clock times than normal, Degraded Low means shorter clock times than normal.

did occur, a new baseline is identified based on identified “good” data. After the baseline is identified, outliers (assumed to be related to inscription errors) are removed based on a +/- 8% variation threshold from the baseline. This helps to eliminate a good portion of transcription errors but not all.

The data used in this study was broken into training sets of known Launch Valve performance data and two test sets of unknown performance data (known by SME supplying the data). Specifically, the training sets contained 27,622 sequential shots of healthy performance data and 11,882 sequential shots that contained degraded performance within the data set. Test Set 1 contained 19,355 sequential shots and Test Set 2 contained 10,648 sequential shots.

4.2. Feature Extraction

The prognostics algorithm presented in this work, starts with the assumption that the following data has been

received: shot number, Clock 2 times, and base line times for all catapult shots. The extraction of these times was described in the previous section. To account for any shift in the Clock 2 timing of the valve, the clock times ($time_{obs}$), are normalized using the baseline time ($time_{exp}$), resulting in $\boldsymbol{Clock\ 2\ Data}$ as illustrated in Eq. (1).

$$\boldsymbol{Clock\ 2\ Data} = \frac{time_{obs} - time_{exp}}{time_{exp}} \qquad (1)$$

The algorithm tracks all aircraft shots. Both BTNLs and aircraft shots are used to track performance. Figure 3 shows an example set of Clock 2 data of a healthy Launch Valve over a one year period.

The distribution of the Clock 2 data, C, over N launch cycles, $p_N(C)$, data tends to fit a Gaussian distribution of the following form:

$$p_N(C) \rightarrow \frac{1}{\sigma\sqrt{2N\pi}} e^{-\frac{(C-N\mu)^2}{2N\sigma^2}} \qquad (2)$$

which is the formula for a Gaussian distribution with mean $N\mu$ and variance $N\sigma^2$.

Based on consultations with SMEs, it was determined that degraded operation resulted in a shift of the mean and a change in the standard deviation of the clock times. There are two different degraded modes. Data that has an increasing mean (slower clock times, Degraded High) can be representative of a valve experiencing high internal friction; while data that has a decreasing mean (faster clock times, Degraded Low) can be representative of a valve leakage in hydraulic fluid downstream. An example of this is demonstrated in Figure 4 where the blue data (solid line) represents a healthy Launch Valve and the red data (dots/dashes) represents a valve operating in a degraded condition (low – dashed line, high – dotted line). These distribution functions were extracted by analyzing the training set of known healthy, degraded low, and degraded high valve performance data. The mean and standard deviation are used as features to detect anomalies in the clock data.

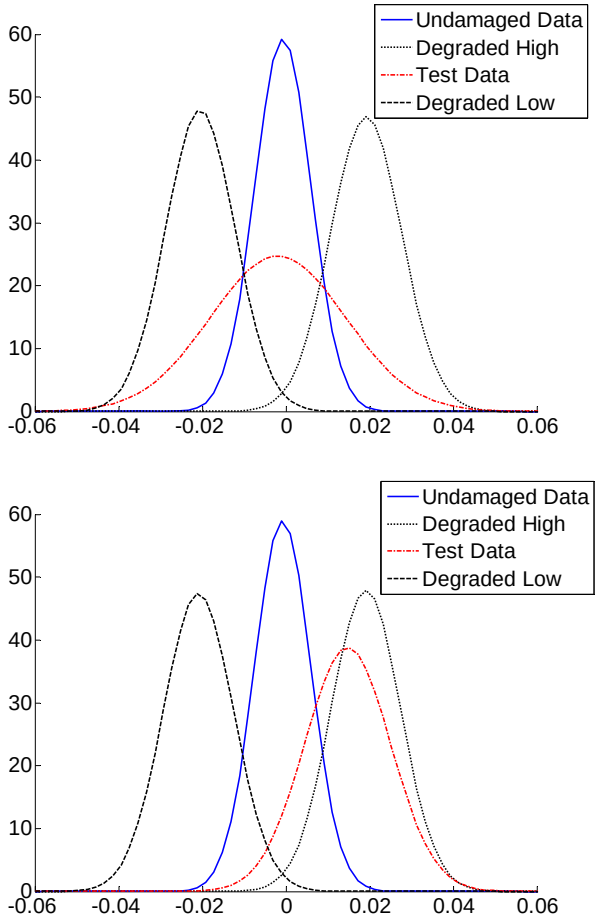


Figure 5: PDFs of performance data. Top) Test data is still in the good performance range. Bottom) Test data has shifted into the degraded high range.

4.3. Anomaly Detection

This work implements a data driven approach for detection of degradation in Launch Valve performance. The problem simplifies to an anomaly detection problem, i.e. detecting when the incoming signal (features) are diverging from a historically estimated healthy state. Parameters for the healthy state are extracted from a known healthy training set of data and used in the comparison against incoming data. A hypothesis test is conducted using the Neyman-Pearson Criterion (Lehmann 1986). Neyman-Pearson is a probabilistic method used to classify data points in a null or alternative hypothesis by calculating a likelihood ratio and comparing it to a threshold.

In the case of the Launch Valve, the two different degraded modes lead to two alternative hypotheses, Degraded High or Degraded Low. Table 1 shows the designation of these states.

Table 1: Neyman-Pearson Hypotheses

H_0	Null Hypothesis that the Launch Valve is healthy
$H_{1_{High}}$	Hypothesis that the Launch Valve is degraded indicated by slower clock times
$H_{1_{Low}}$	Hypothesis that the Launch Valve is degraded indicated by faster clock times

Figure 5 Top provides a visual representation of the various performance distributions. The black probability distribution functions (PDFs) represent degraded low and high data, the blue PDF is an undamaged set of data, and the red PDF is an example set of test data. The increased standard deviation in the test data may be due to intermittent inconsistencies in lubrication during operation. The Neyman-Pearson Criterion calculates the likelihood ratio, $L(\boldsymbol{x})$ (shown in Eq. 3), which is the ratio of the probability of a data set belonging to the alternative hypothesis versus the null hypothesis. The probability of accepting $H_{1_{High}}$ increases when the test dataset starts to shift, as seen in Figure 5 Bottom.

NOTE: For future reference, any degraded state will be represented by H_1 unless a low/high degraded state is specifically stated.

Two false alarm rates, Type I Error and Type II Error, must be specified to correctly classify an anomaly. Table 2 below shows the designations of both of these errors.

Table 2: False Alarm Rate Designation

P_{FA_I}	Probability of Type I Error (False Positive: Conclude damage is present falsely)
$P_{FA_{II}}$	Probability of Type II Error (False Negative: Conclude damage is not present falsely)

The probability of a Type I Error was set to 0.01 yielding a probability of detection of 99%. The likelihood ratio is then calculated to help classify when the measured data set **x** signifies degraded operation. If this ratio is greater than one, there is a higher probability of accepting the alternative hypothesis.

$$L(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|H_1)}{p(\boldsymbol{x}|H_0)} \quad (3)$$

To better utilize the measurement distribution, a window (size W=100 launch cycles) of timing data $\boldsymbol{x}$ is used in the likelihood ratio as follows:

$$L(\boldsymbol{x}) = \frac{\prod_{i=1}^{W} p(x_i|H_1)}{\prod_{i=1}^{W} p(x_i|H_0)} \quad (4)$$

The next phase of anomaly detection implements a Sequential Probability Ratio Test (SPRT). The SPRT evaluates deviations of the actual signal from the expected signal (healthy data) based on distributions instead of a single threshold value to determine if data belongs to a degraded state. SPRT uses the log of the likelihood, L(x), in a sequential analysis. (Wald, 1947). The cumulative log-likelihood is calculated, as seen in Eq. (5), and compared against lower and upper thresholds a and b to determine the next course of action (Table 3). As a new sample becomes available, the observation window shifts, calculating a new likelihood ratio and SPRT value.

$$SPRT_i = SPRT_{i-1} + \log\big(L(\boldsymbol{x_i})\big) \quad (5)$$

Table 3: SPRT Comparison Statements

$a < SPRT_i < b$	Continue monitoring
$SPRT_i \geq b$	Accept H_1
$SPRT_i \leq a$	Accept H_0

With a set probability of 1% for a Type I Error and a set probability of 5% for a Type II Error, thresholds a and b are calculated using Eq. (6) and Eq. (7) respectively.

$$a = \ln\left(\frac{P_{FA_{II}}}{1 - P_{FA_I}}\right) = -2.99 \quad (6)$$

$$b = \ln\left(\frac{1 - P_{FA_{II}}}{P_{FA_I}}\right) = 4.55 \quad (7)$$

H_1 is accepted when the SPRT calculation exceeds the b threshold. This concludes there is enough data to support the decision to determine an anomaly has been detected. The SPRT is then reset if the value has declined consecutively for 20 iterations. If H_0 is accepted, the cumulative log-likelihood ($SPRT_i$) is reset to zero to restore sensitivity to small changes in degradation. A similar approach to anomaly detection was implemented by Cheng et. al. for monitoring environmental and operational stress profiles of robotic vehicles (Cheng, 2008).

4.4. Degradation Model

A third order polynomial was chosen as a data-driven damage progression model based on a best fit of multiple degradation sections from the training sets. SMEs also helped to define the ranges for initial parameter distributions for the model parameters based on their experiences with historical performance degradation trends. The performance degradation model follows Eq. (8) where $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are model coefficients, $\boldsymbol{T}$ is the translation parameter allowing the model to adapt to shifting states of degradation, $\boldsymbol{y}$ is the degraded state prediction of the next shot, i is the sample index (with index 1 being the detected start of degradation), and dt is the cycle increment which was set to 1 (each shot increments by 1).

$$\boldsymbol{y}_i = \boldsymbol{a}(i + dt + \boldsymbol{T})^3 + \boldsymbol{b}(i + dt + \boldsymbol{T})^2 + \boldsymbol{c}(i + dt + \boldsymbol{T}) \quad (8)$$

Parameters $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{T}$ are initialized after an anomaly is detected and are updated via the particle filter (described in the next section) for as long as the data classifies the Launch Valve operation as degraded.

The effect of loading conditions (varying aircraft weights) on the degradation of the launch valve performance is negligible. The CSV controls the launch valve rate of opening regardless of what aircraft is on the catapult. In other words, regardless of the aircraft type, if a value of CSV 200 is used to launch a F/A-18 or an EA-6B aircraft (two different weight aircraft), the launch valve clock time should be the same.

4.5. Prediction

Once an anomaly is detected, a particle filtering (PF) based prognostic algorithm takes over. PF prognostic algorithms have become a common method in the state of the art prognostics. A PF is used to provide estimations of distributions of model parameters using a window of observations. This is accomplished using Bayesian inference, based on Bayes' Theorem as seen in Eq. (9), where $\boldsymbol{\Theta}$ is a vector of unknown parameters (a,b,c,T), $p(\boldsymbol{\Theta})$is the prior PDF of these parameters, $\boldsymbol{z}$ is the vector of observed data (clock 2 time), $p(\boldsymbol{\Theta}|\boldsymbol{z})$ is the posterior PDF of $\boldsymbol{\Theta}$ conditional on $\boldsymbol{z}$ and $L(\boldsymbol{z}|\boldsymbol{\Theta})$ is the likelihood of the observed data given the parameters (An, 2012).

$$p(\boldsymbol{\Theta}|\boldsymbol{z}) \propto L(\boldsymbol{z}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}) \qquad (9)$$

The particle filter utilizes a sequential method of passing prior estimations into the current step to produce the estimations for the next step. In particular, this work implements a simplified version of the Risk-Sensitive

Particle Filter (RSPF) presented by Orchard et. al. (Orchard, 2010). The RSPF maintains a subset of particles in the high-risk, low-likelihood realm to maintain coverage in these areas when incoming data causes convergence of particles to a single particle or narrow distribution. In this work, twenty percent of the particles are allocated to maintain distribution within the risk sensitive areas.

Input into the PF is timing data that has been filtered with two passes of an exponential moving average filter (EMAF) as shown in Eq. (10). Development with training data supported using parameters $\alpha = 0.003$ on the first pass and 0.03 on the second pass. The EMAF is an infinite impulse response discrete filter that provides low latency.

$$EMAF_i = \alpha f_i + (1-\alpha)EMAF_{i-1} \qquad (10)$$

The degradation model parameters are estimated using a 10 sample window of EMAF data. Using a sample from the EMAF data, a likelihood calculation is performed and 1000 particle weights are updated. Each particle represents a particular parameter configuration with a particle weight

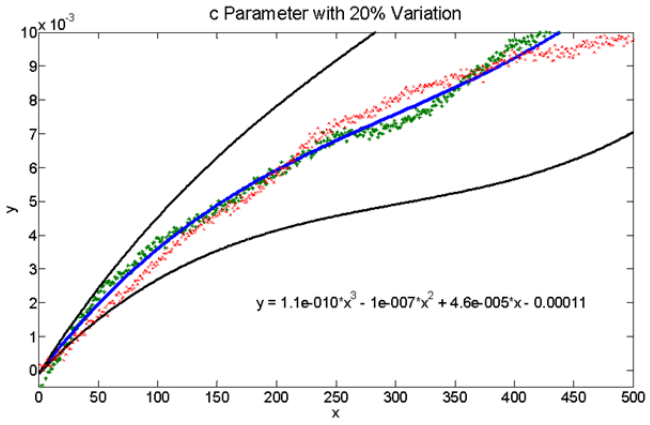


Figure 6: Performance degradation plots. Two examples showed (darker dots, lighter dots). Third order model fit to data. 20% Bounds on c parameter shown by black lines.

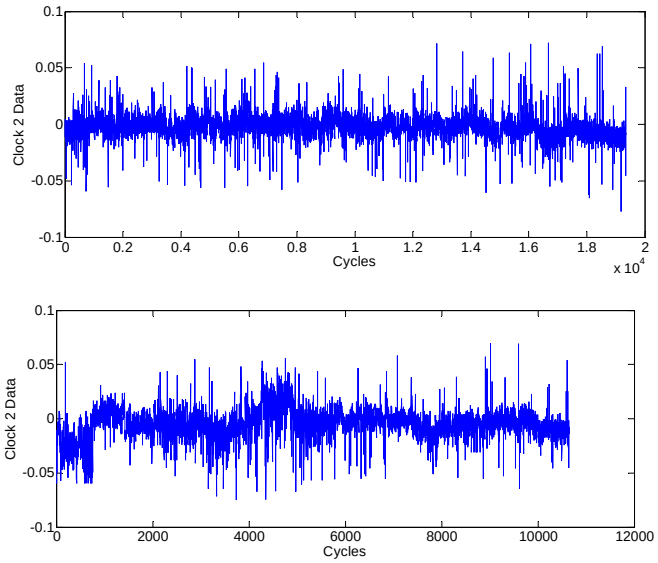


Figure 7: Top) Test set 1. The algorithm classified this test set as containing all healthy data, Bottom) Test set 2. The algorithm classified this test set as containing degraded performance data.

based on its likelihood. These weights are then used in the likelihood calculation for the next measurement sample of the current EMAF window. Parameters are updated for each sample of the window and the resulting particle weights are used in a third order model to generate each particle prediction.

Once predictions have exceeded the failure threshold (defined by the SME), each particle contributes to the time until MIR PDF. When a new measurement data point is acquired, the EMAF output is updated and the particle filter

window is shifted. The shifted window is then passed through the process to update the parameter weights and provide a prognosis, utilizing a portion of the weights from the previous measurement. The prognosis process repeats, resulting in updated MIR predictions as the degradation progresses.

5. RESULTS

The algorithm was tested against two sets of data, shown in Figure 7, of unknown classification to the program (but known by the SME who supplied the test sets). For each classification test, the algorithm was fed the test data cycle by cycle, as if it was being deployed in real time. Once the observation window is filled, each data point was classified as belonging to a degraded state or a healthy state. Overall the test sets were classified as "healthy" if they had no anomaly detections and "degraded" if anomalies were detected. The algorithm classified Test Set 1 as containing only healthy data and Test Set 2 as containing degraded performance data. The SME validated that this was the correct classification for the data that he supplied. Furthermore, for Test Set 2, the algorithm identified locations in time for which degraded performance was identified (shown in Figure 8).

At the start of identified degradation (rising edge on plot in Figure 8), the prediction algorithm took over and predicted out when the performance data would cross a pre-defined "maintenance needed" threshold. An example is shown in Figure 9 where an anomaly was detected around cycle shot 4290 and predictions were made for the remaining cycles until maintenance would be required. The figure shows an example of predictions to MIR at about 50% remaining time until MIR.

To assess the quality of the prediction for Test Set 2 (shown in Figure 9), the Alpha-Lambda performance metric is used (Saxena 2009). The Alpha-Lambda performance metric is an off-line metric that determines whether the prediction falls within the specified levels of a performance measure at particular times. The time instances are specified as a percentage of total remaining life (cycles until MIR in this case) from the point the first prediction is made. Accuracy, defined as the prediction accuracy of cycles until MIR, is set to be alpha*100% of the actual cycles until MIR. In this case, an alpha of 0.2 was used. Results from Test Set 2 consistently showed the prediction of remaining cycles until MIR fell within the 20% accuracy (alpha = 0.2) with approximately 70% (lambda = 0.7) of the remaining cycles until MIR remaining. This can be seen in Figure 10. Early predictions in the normalized prognostic window tend to fluctuate outside the Alpha-Lambda cone due to wide spread in the distribution of particles used in the particle filter. As more degraded data is acquired, the particle distribution tightens as the particle filter begins to converge on a particular degradation model.

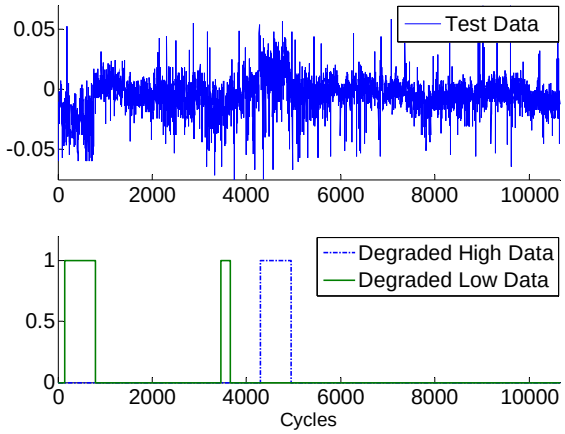


Figure 8: Test set 2 with algorithm identified locations with degraded performance in both low (green) and high (blue) levels. "Low" means timing is shorter than normal, "High" means timing is longer than normal.

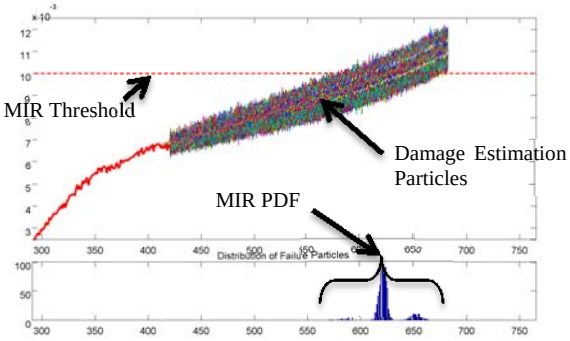


Figure 9: Particle Filter Estimation of degradation and MIR PDFs.

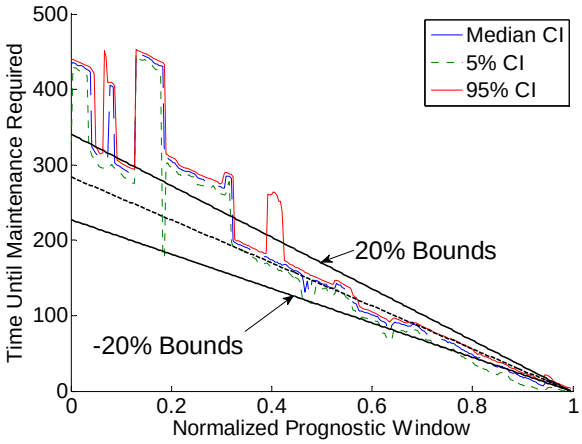


Figure 10: Alpha-Lambda Performance with 20% error bound. Prediction until MIR showing Median, 5%, and 95% confidence levels (Cl).

6. CONCLUSIONS AND FUTURE WORK

For the Low Loss Launch Valve, the method of extracting and using features from timing data such, as mean and standard deviation, to detect anomalies using Neyman-Pearson Theorem and SPRT has been shown in the previous section to produce promising results. The prediction of the remaining time until MIR with a risk sensitive particle filter using a third order model has also been shown to produce results within an acceptable accuracy window. This is a step towards allowing the Launch Valve performance analysis to be handled automatically in real-time onboard ship and provide timely status information to the fleet.

The next step toward achieving an automated PHM solution for the Low Loss Launch Valve is to automate the process of pre-scrubbing the data which is currently handled by the SME. The automated pre-scrub would need to receive raw

clock timer information (CSV setting and Clock 2 time), screen out low CSV launches not useable for review, and properly identify baseline shifts without input from users. The algorithm needs to handle varying levels of noise / error in the data, much due to transcription errors. It is possible that future upgrades to the launch system could incorporate added sensors and electronic logging to automatically record the timing data, thereby eliminating transcription error issues.

Acquiring more test data sets would further verify / validate the PHM methodology presented in this work. With more data, it is possible that supervised learning algorithms such as neural networks could be used to improve upon classification methods and anomaly detection. Future work will also include methods of identifying healthy data in real-time data sets (deployed system) and use that to set anomaly detection and prognostics parameters. This would reduce reliance on fleet historical data and would tailor PHM methods to each specific Launch Valve system through its life span.

REFERENCES

An, D., Choi, J., Kim, N.H., (2012), "A Tutorial for Model-based Prognostics Algorithms based on Matlab Code," *Annual Conference of Prognostics and Health Management Society*, vol. 3, no. 122, September 2012.

Cheng, S. (2008), Autonomous Prognostic Monitoring Device. *Proceeding for 62nd Meeting of the Society for Machinery Failure Prevention Technology (MFPT)*, Virginia Beach, VA, May 2008

Daigle, M., Goebel, K. (2011), "A model-based prognostics approach applied to pneumatic valves," *International Journal of the PHM Society*, vol. 2, no. 2, pp. 1-16.

Gomes, J., Ferreira, B., Cabral, D., Glavao, R., Yoneyama, T. (2010), "Health Monitoring of a Pneumatic Valve Using a PIT Based Technique," *Annual Conference of Prognostics and Health Management Society*

E. Lehmann (1986), *Testing Statistical Hypotheses*, New York: Wiley, 1986.

Orchard, M., Tang, L., Saha, B., Goebel, K., Vachtsevanos, G. (2010), "Risk-Sensitive Particle-Filtering-based Prognosis Framework for Estimation of Remaining Useful Life in Energy Storage Devices," *Studies in Informatics and Control*, vol. 19, no. 3, September 2010.

Saxena, A; Celaya, J.; Saha, B.; Saha, S.; Goebel, K. (2009), "Evaluating Algorithm Performance Metrics Tailored for Prognostics," *IEEE Aerospace Conference*, vol., no., pp.1,13, 7-14 March 2009.

Wald, A. (1947), "Sequential Analysis", *John Wiley & Sons*, New York, NY,.

BIOGRAPHIES

Glenn Shevach is an Electrical Engineer in the Advanced Technology Projects Branch Code 4.8.1.4. at NAWCAD in Lakehurst, NJ. He holds a Masters Degree in Electrical Engineering from Stevens Institute of Technology (2010). He is the Advanced Arresting Gear S&T / Prognostics Lead.

Mark Blair is a Computer Engineer in the Advanced Technology Projects Branch Code 4.8.1.4. at NAWCAD in Lakehurst, NJ. He is currently working towards a Masters Degree in Systems Engineering from Stevens Institute. His current work focuses on S&T related topics for Aircraft Launch Recovery and Support Equipment (ALRE & SE).

James Hing is a Mechanical Engineer in the Advanced Technology Projects Branch Code 4.8.1.4 at NAWCAD in Lakehurst, NJ. He holds a PhD in Mechanical Engineering from Drexel University (2010). His current work focuses on robotics and prognostics solutions for ALRE & SE applications.

Larry Venetsky is an Electrical Engineer in the Advanced Technologies Project Branch Code 4.8.1.4 at NAWCAD in Lakehurst, NJ. He holds MS in Electrical Engineering from Drexel University (1993). His current work focuses on robotics and machine learning for ALRE & SE applications.

Everard Martin is a Mechanical Engineer in the Steam Catapult Launcher In-Service Branch, Code 4.8.2.1 at NAWCAD Lakehurst, NJ. He holds a BSE in Mechanical Engineering from Drexel University (1981). He has been working Steam Catapult Performance and Certification since June 1989.

John Wheelock is a Mechanical Engineer in the Steam Catapult Launcher In-Service Branch, Code 4.8.2.1 at NAWCAD Lakehurst, NJ. He holds a BSE in Mechanical Engineering from Arizona State University (2002). He has been working Steam Catapult Performance and Certification with a primary focus on catapult performance monitoring since July, 2002.

Prognostic Modeling of Valve Degradation within Power Stations

M. J. McGhee [1], G. Galloway[1], V. M. Catterson[1], B. Brown[2] and E. Harrison[2]

[1] *University of Strathclyde, Glasgow, United Kingdom*
mark.j.mcghee@strath.ac.uk
grant.galloway.2013@strath.ac.uk
v.m.catterson@strath.ac.uk

[2]*GSE Systems, Glasgow, United Kingdom*
blair.brown@gses.com
emma.harrison@gses.com

ABSTRACT

Within the field of power generation, aging assets and a desire for improved maintenance decision-making tools have led to growing interest in asset prognostics. Valve failures can account for 7% or more of mechanical failures, and since a conventional power station will contain many hundreds of valves, this represents a significant asset base. This paper presents a prognostic approach for estimating the remaining useful life (RUL) of valves experiencing degradation, utilizing a similarity-based method. Case study data is generated through simulation of valves within a 400MW Combined Cycle Gas Turbine power station. High fidelity industrial simulators are often produced for operator training, to allow personnel to experience fault procedures and take corrective action in a safe, simulation environment, without endangering staff or equipment. This work repurposes such a high fidelity simulator to generate the type of condition monitoring data which would be produced in the presence of a fault. A first principles model of valve degradation was used to generate multiple run-to-failure events, at different degradation rates. The associated parameter data was collected to generate a library of failure cases. This set of cases was partitioned into training and test sets for prognostic modeling and the similarity based prognostic technique applied to calculate RUL. Results are presented of the technique's accuracy, and conclusions are drawn about the applicability of the technique to this domain.

Mark McGhee et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Within electrical power utilities there is an increasing demand for condition monitoring methods capable of reliably predicting the RUL of assets (Sheppard & Kaufman 2009). This requirement is driven by the need to improve maintenance costs and scheduling, as well as safety considerations (Chen, Yang & Zheng 2012). The field of prognostics has made great advances in areas with high requirements on safety and dependability, such as aerospace and the nuclear industry. However within the power generation field, prognostic applications have not been implemented to the same degree. This is mainly due to the challenges of gathering sufficient data to enable robust testing and validation, as such systems are rarely allowed to run to failure (Heng, Tan, Mathew, Montgomery, Banjevic, & Jardine, 2009).

Within power generation, implementation of prognostic methods would enable operators to reduce maintenance and unplanned downtime by utilizing predictive maintenance policies in place of a time based maintenance approach (Vachtsevanos, Lewis, Roemer, Hess & Wu, 2006) (Sun, Zeng, Kang & Pecht 2012). However, there is a high cost associated with creating physical test systems from which to gather run-to-failure data. Additionally, gathering, understanding, and transforming data provided by on-site industrial facilities into a comprehensive and reliable model is a costly and difficult undertaking (Wenbin & Carr 2010), with operators often reluctant to provide commercially sensitive data.

One way to overcome this lack of failure data is to utilize simulation of assets to generate the data required. Following

this route, this paper proposes the simulation of degradation of valves within a power plant environment to create a similarity-based prognostic model. Within a plant environment, valves have been highlighted as a common source of faults, accounting for at least 7% of mechanical failures (Radu, Mladin & Prisecaru, 2013) (Latcovich, Åstrom, Frankhuizen, Fukushima, Hamberg & Keller, 2005), and with many hundreds of valves present in a typical generation plant (Westinghouse Nuclear, 2013), valves are a critical asset which could benefit from a prognostic system.

Within power generation, simulators have been widely deployed, particularly within the nuclear sector, for training purposes focused on improving operational safety (Harrison, 2013). Such simulators are used primarily for training and are certified as high fidelity tools and thereby the model and sensor data are within industrially accepted tolerances of actual plant values. Utilizing such high fidelity simulators negates the need for the creation of physical test beds, as well as providing an industrial acceptance and robustness to the simulated data generated (McGhee, Catterson, McArthur and Harrison, 2013).

The similarity-based prognostic method used here is based on an approach by Wang, Yu Siegel and Lee (2008). This similarity method has particular application benefits to the simulation approach proposed here. With simulation, the large number of run-to-failure cases needed for a similarity based approach can be generated easily. The use of simulation can also satisfy the requirements stated by Wang et al. (2008) for a successful implementation:

1) Multiple recordings of run-to-failure data are available,

2) The data recorded ends when the point of failure is reached, and

3) The data covers a representative set of components.

2. METHODOLOGY

This section discusses the creation of the valve failure model and the prognostic RUL model. A diagram of the process is shown in Figure 1.

2.1. Valve model simulation

The valve model was created from first principles, simulating fluid flow within a cylindrical pipe:

$$P_2 = P_1 + \frac{1}{2}\rho(V_1^2 - V_2^2) \quad (1)$$

$$A_1V_1 = A_2V_2 \quad (2)$$

Where P_1, V_1 and A_1 correspond to the pressure, fluid flow and area of the pipe entering the valve, P_2, V_2 and A_2 correspond to the pressure, fluid flow and area of the pipe at the point of degradation and ρ describes the density of the fluid. Parameter values for the model are taken from an industrial Combined Cycle Gas Turbine (CCGT) plant simulator.

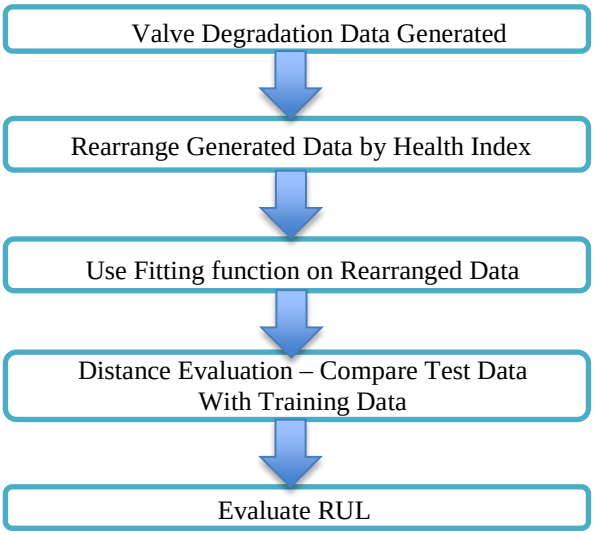


Figure 1. Procedure of RUL estimation

The degradation is represented by a decreasing area A_2 where the initial area of the pipe A_1 is constricted over time. This is represented by a degradation coefficient, δ, which is a numerical constant between 0 and 0.0001, drawn from a standard uniform distribution, describing the rate of decrease in the flow area.

$$A_2(t+1) = A_1(0) - \delta A_1(t) \quad (3)$$

This degradation can represent debris build up along the area of flow, or "sticky valve failure" where the valve no longer fully closes or opens. A single run-to-failure event from initial healthy operating conditions to end of life can be seen in Figure 2, and a batch of 50 run-to-failure events can be seen in Figure 3. For this study, the end of life is considered to be $P_2 = 0$, i.e. completely blocked flow. However, in a power station deployment, maintenance intervention would be triggered significantly before this threshold is reached.

This modeling approach corresponds to the way components and faults are modeled in the industrial plant simulator used in the research. The plant simulator uses first principles equations based on pressure, fluid flow and flow area to model pipes and valves.

The modeling choices also need to be made with respect to the sensors and data readily available to station operators. Theoretically, measurement points could be placed at any point in the plant model, and the parameter value recorded

as if from instrumentation. However, for the prognostic model to translate directly from the plant simulator to the real plant environment, any measurements utilized by the prognostic model must be realistic points for instrumentation to be located. Therefore, only those parameters which would normally be recorded around a valve are considered.

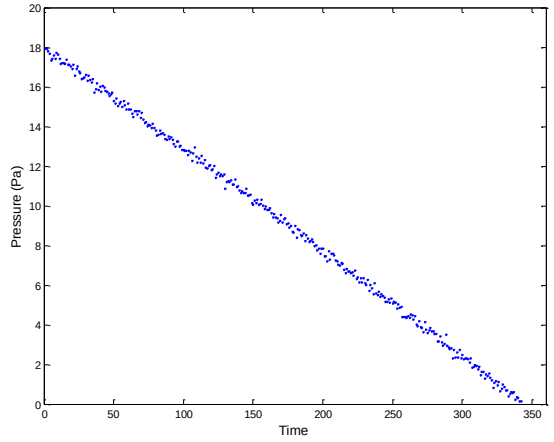


Figure 2. A single run-to-failure event

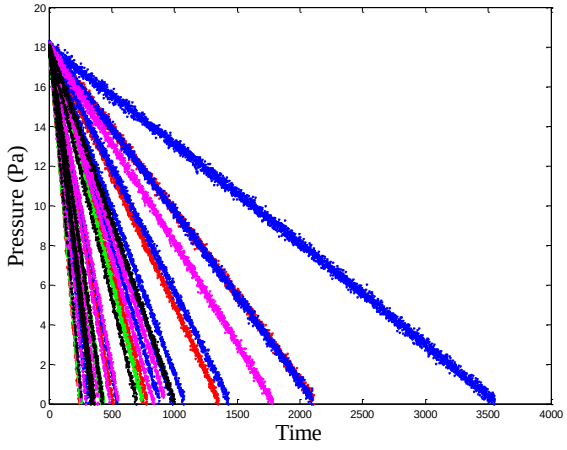


Figure 3. 50 run-to-failure events

For this study, the training data comprised 50 sets of time stamped pressure values, corresponding to P_2 in Eq. (1), from an initial value equal to P_1 down to 0. The simulated frequency of data capture is set at once per hour. For this case, the parameters taken from the CCGT were an initial pressure P_1=18 Pa, area A_1=10 cm^2 and flow V_1=185kg/s. To represent measurement noise, each data point had a noise term added, drawn from a Gaussian distribution with mean 0 and standard deviation 0.0005.

2.2. Prognostic model

The procedure for creating the similarity-based prognostic model is split into three steps (Wang et al., 2008). The first two, described in sections 2.2.1 and 2.2.2, are data preparation steps applied to both training and test data. The third step compares the test data set against the training data. Of 55 run-to-failure events simulated, 50 were used as training data, with five for testing.

2.2.1. Arrangement by health index

The initial stage is to rearrange the data to create a Health Index (HI). The HI is used to describe the condition of the asset. Near the start of life the asset is assumed to be in a healthy condition and assigned the value 1, whilst the unhealthy or near end-of-life condition is assigned the value 0. This HI is then applied to every data run and the data rearranged according to the asset's time-to-failure (Figure 4). As shown in Figure 4, the start of life (healthy) and end of life (unhealthy) values correspond to P=18 and P=0 respectively.

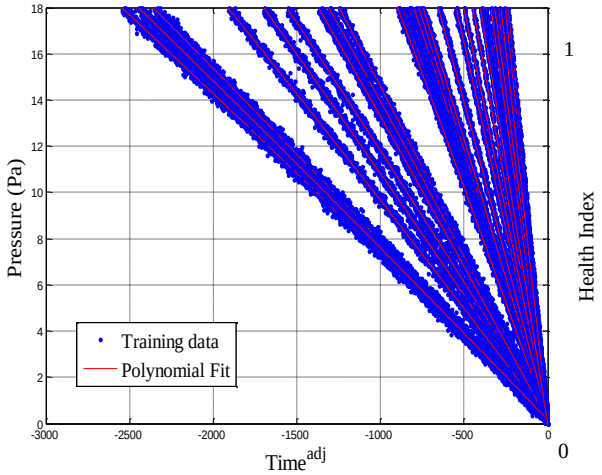


Figure 4. Training set comprising 50 run-to-failure events rearranged according to HI

Polynomial fitting

Having rearranged the data according to the HI, each run-to-failure event is then fitted using a polynomial function which best describes the event progress. In the specific case of this valve degradation example, the fault progression looks to approximate a linear fit. However, in other cases the best fit may be a higher order polynomial or other function. In this case the polynomial fit is:

$$f(x) = ax + b \quad (4)$$

where a and b are the model parameters. This polynomial curve is fitted to the HI for every run-to-failure event with the least squares fitting approach.

2.2.2. Distance Evaluation

To determine the RUL of the test runs, a sample of data from near the start of each test is selected. In the examples below, time steps 50–100 are chosen to represent the current and recent historic condition of the valve. This data is then compared against every 50 time step segment of each training data polynomial fit until the closest match to the test is found. The distance evaluation is determined by:

$$d(\tau, Y, i) = \sum_{j=1}^{r} \frac{(y_j - f_i(-\tau - r + j))^2}{\sigma_i^2} \qquad (5)$$

where d is the distance of the test data from the training data sample, y is the position of the test data (time step number), f_i is the polynomial curve fitted to the ith training data sample, r is the length of the test data Y, τ is the number of time steps Y is shifted from 0 and σ is the RMS error from the polynomial fit.

Once the distance between the test run and all windows of all training runs is established, the estimated RUL is chosen by selecting the training run sample with the smallest distance d (i.e. the most similar run-to-failure event). The RUL from that point of the training run is the estimated RUL for the test run.

3. EXPERIMENTAL RESULTS

The five test runs are summarized in Table 1 and shown in Figures 5 – 9. As can be seen, the true RUL of each test run compares well with the predicted RUL value.

Table 1. Summary of Test run results with associated Estimated RUL and True RUL

Test Run	*Est RUL*	*True RUL*
1	230	239
2	898	889
3	631	624
4	673	638
5	1204	1195

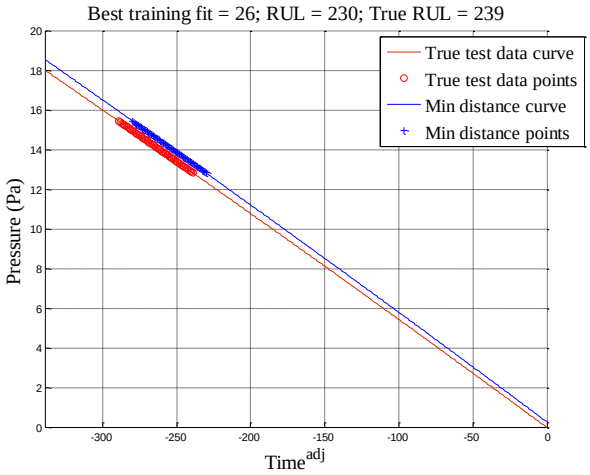


Figure 5. Test run 1: Estimated RUL = 230, True RUL = 239

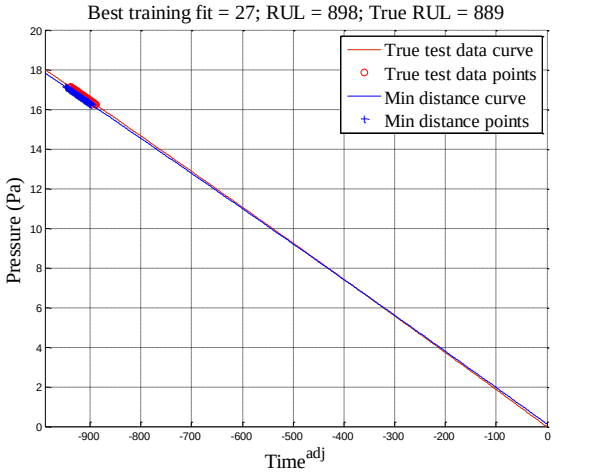


Figure 6. Test run 2: Estimated RUL = 898, True RUL = 889

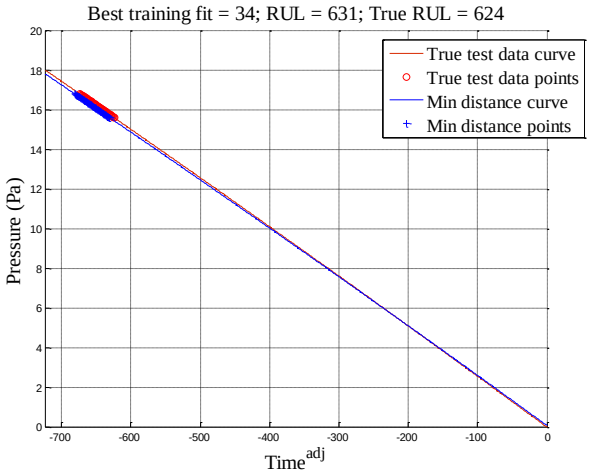


Figure 7. Test run 3: Estimated RUL = 631, True RUL = 624

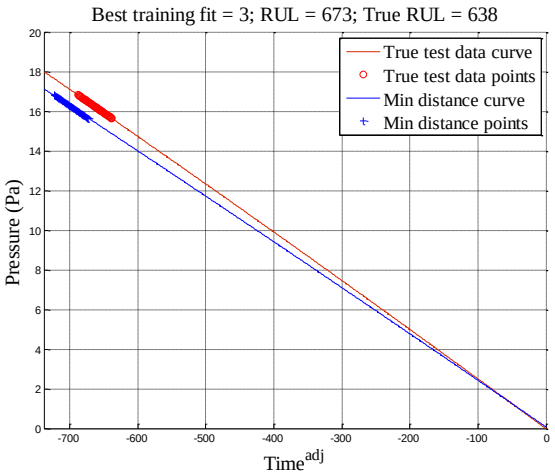


Figure 8. Test run 4: Estimated RUL = 673, True RUL = 638

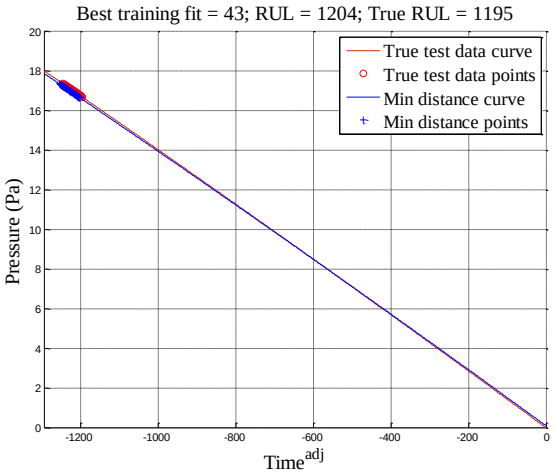


Figure 9. Test run 5: Estimated RUL = 1204, True RUL = 1195

These results are considered accurate enough for the application domain, being within 10 hours of the actual RUL in most cases, and 35 hours in the worst case. While this technique estimates the time to complete failure (zero flow), in a power station maintenance would be triggered by a reduction in flow, significantly before failure. The estimation of RUL gives an indicative window of time in which maintenance could or should be performed, thus providing support to maintenance planning. Future work will consider how far in advance of estimated failure a maintenance trigger should be set, bearing in mind uncertainties in the RUL prediction.

The high accuracy of the case study RUL predictions is due to the range of failures included in the training data set, which is due in turn to the use of simulation. With the high fidelity plant simulator, plant conditions can be varied and reset for multiple fault runs, generating as many failure examples as desired.

There is potential for this similarity based prognostic method to be improved further, with a larger training data set containing a greater breadth of degradation and failure cases. Future work will consider how large the training set needs to be, and how to integrate actual valve failure data as it becomes available.

However, as more training data is added, RUL selection becomes more complex. Future extensions of this technique may need to consider implementing different methods of distance evaluation, to retain prediction accuracy. Also, as this method relies on training using run-to-failure data, it is limited to accurate prediction of previously seen fault types.

4. CONCLUSIONS

The similarity-based prognostic approach described in this paper provided accurate results when estimating RUL of valves within a power station. This research utilizes a high fidelity CCGT plant simulator to allow the creation of a large suite of failure cases, simulating a relatively low risk but high consequence failure mode for which there is limited in-service data. This paper demonstrates a method of first principles modeling of failure, in order to generate the data required for data-driven prognostic modeling. This is shown to accurately predict the remaining life of five test cases.

Having tested the method there are a number of possible routes now available for further research using this approach: testing the approach with real plant data, applying the prognostic method to different types of faults, and comparing this technique to other prognostic techniques for similar applications.

ACKNOWLEDGMENTS

The authors would like to thank GSE Systems for the use of their high fidelity simulation suite and technical support during this research.

REFERENCES

Chen, Z.S., Yang, Y.M. & Zheng Hu, (2012) A Technical Framework and Roadmap of Embedded Diagnostics and Prognostics for Complex Mechanical Systems in Prognostics and Health Management Systems, *IEEE Transactions on Reliability*, Vol. 61, (Issue: 2), Pages: 314 – 322, doi: 10.1109/TR.2012.2196171

Harrison, S. (2013), The Case for Simulation and Visualisation Based Training, *Marine Electrical and Control Systems Safety Conference, (MECSS 2013)*, October 2-3, Amsterdam

Heng, A., Tan, A. C. C., Mathew, J., Montgomery, N, Banjevic, D. & Jardine, A. K. S., (2009), Intelligent Condition-Based Prediction of Machinery Reliability,

Mechanical Systems and Signal Processing, Vol. 23, (Issue 5), Pages: 1600 – 1614, doi: 10.1016/j.ymssp.2008.12.006

Latcovich J., Åstrom T., Frankhuizen P., Fukushima, S.,Hamberg H., & Keller, S., (2005), Maintenance and Overhaul of Steam Turbines, *International Association of Engineering Insurers 38th Annual Conference*, September, Moscow

McGhee M. J., Catterson V.M., McArthur S.D.J. & Harrison E. (2013), Using a High Fidelity CCGT Simulator for building Prognostic Systems, *European Technology Conference 2013. EuroTechCon 2013*, November 19-21, Glasgow, UK

Radu G., Mladin D. & Prisecaru I. (2013) Analysis of potential common cause failure events for Romania-TRIGA 14 MW reactor, *Nuclear Engineering and Design*, Vol. 265, Pages: 164-173, doi: 10.1016/j.nucengdes.2013.06.027

Sheppard, J.W., Kaufman, M.A. & Wilmer, T.J. (2009), IEEE Standards for Prognostics and Health Management, *IEEE Aerospace and Electronic Systems Magazine*, Vol. 24, (Issue: 9), Pages: 34 – 41, doi: 10.1109/MAES.2009.5282287

Sun, B., Zeng, S., Kang, R. & Pecht, M.G. (2012) Benefits and Challenges of System Prognostics, *IEEE Transactions on Reliability*, Vol. 61, (Issue: 2), Pages: 323 – 335, doi: 10.1109/TR.2012.2194173

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering system*. Hoboken, NJ: John Wiley & Sons, Inc

Wang, T., Yu J., Siegel D. & Lee J. (2008). A Similarity Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems, *International Conference on Prognostics and Health Management, 2008. PHM 2008*, October 6-9, Denver, CO, 10.1109/PHM.2008.4711421

Wenbin Wang & Carr, M.,(2010), A Stochastic Filtering Based Data Driven Approach for Residual Life prediction and Condition Based Maintenance Decision Making Support, *Prognostics and Health Management Conference, 2010. PHM '10*, Jan 12-14, Macao, 10.1109/PHM.2010.5413485

Westinghouse Nuclear, (2013), AP1000 PWR Nuclear Reactor Brochure

BIOGRAPHIES

Mark J. McGhee is a PhD student within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his MSci in Applied Physics from the University of Strathclyde in 2012. His PhD focuses on condition monitoring and prognostics for power plant systems, in collaboration with GSE Systems, a leading provider of high fidelity industrial simulation technology and training solutions.

Grant S. Galloway is a PhD student within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his M.Eng in Electronic and Electrical Engineering from the University of Strathclyde in 2013. His PhD focuses on condition monitoring and prognostics for tidal turbines, in collaboration with Andritz Hydro Hammerfest, a leading tidal turbine manufacturer.

Victoria M. Catterson is a Lecturer within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. She received her B.Eng. (Hons) and Ph.D. degrees from the University of Strathclyde in 2003 and 2007 respectively. Her research interests include condition monitoring, diagnostics, and prognostics for power engineering applications.

Blair Brown is a Simulation Engineer with GSE Systems, Glasgow, UK.

Emma Harrison is Business Projects Director with GSE Systems, Glasgow, UK.

Validation of Model-Based Prognostics for Pneumatic Valves in a Demonstration Testbed

Chetan S. Kulkarni[1], Matthew Daigle[2], George Gorospe[1], and Kai Goebel[2]

[1] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
chetan.s.kulkarni@nasa.gov
george.gorospe@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov
kai.goebel@nasa.gov

ABSTRACT

Pneumatic-actuated valves play an important role in many applications. When valves are critical to the successful operation of the system, prognostics of these valves becomes extremely important and valuable. In order to facilitate the validation of prognostics algorithms for pneumatic valves, we have constructed a pneumatic valve testbed for use with a cryogenic propellant loading system. The testbed enables the injection of faults with a controllable fault progression profile. Specifically, we can introduce controllable pneumatic gas leaks, the most common faults associated with pneumatic valves. We focus on a valve that moves discretely between open and closed position, and is controlled through a solenoid valve. In this paper, we apply a model-based prognostics approach for pneumatic valves on the testbed. We demonstrate the approach using real experimental data obtained from the testbed.

1. INTRODUCTION

Pneumatic-actuated valves play a critical role in many systems. For example, they are used to control the flow of propellant in cryogenic propellant loading systems, and failures can have an adverse impact on system safety and launch availability (Daigle & Goebel, 2011a). This motivates the need for valve health monitoring and prognosis. To facilitate the maturation of prognostics technology, testbeds can be constructed that allow for fault injection with controllable fault progression profiles, which have been developed for electrical power systems (Poll, Patterson-Hine, Camisa, Garcia, et al., 2007; Poll, Patterson-Hine, Camisa, Nishikawa, et al., 2007), electromechanical actuators (Balaban et al., 2010), and mobile robots (Tang, Hettler, Zhang, & DeCastro, 2011; Balaban et al., 2013). For the purpose of maturing and validating valve prognostics approaches, we have developed a pneumatic valve testbed (Kulkarni, Daigle, & Goebel, 2013).

Whereas earlier work on valve prognosis used algorithms centered on particle filters (Daigle & Goebel, 2011a, 2011b, 2010), in this paper we use a new model-based method based on the measurement of valve open and close times, recently developed in (Daigle, Kulkarni, & Gorospe, 2014). In real valve operations, typically only valve position is measured, from which the only meaningful information for prognostics are the valve open and close times. The new approach is therefore much simpler and requires significantly less computation to isolate and identify faults, and predict end of life (EOL) and remaining useful life (RUL). The approach still follows the general estimation-prediction framework developed in the literature for model-based prognostics (Orchard & Vachtsevanos, 2009; Daigle & Goebel, 2013). In (Daigle et al., 2014), the approach was demonstrated in simulation; in this paper, we apply the approach using real data from the pneumatic valve testbed.

The structure of the paper is as follows. Section 2 discusses the overall setup of the valve prognostics testbed. Section 3 presents the valve model. Section 4 provides the valve prognosis framework, and Section 5 presents prognosis results using testbed data. Section 6 concludes the paper.

2. VALVE TESTBED

The valve prognostics testbed, shown in Fig. 1, has been developed to demonstrate valve prognosis in the context of cryogenic refueling operations (Kulkarni et al., 2013). The dashed lines denote the electrical signals, including the data acquisition I/O signals, power lines, etc. The solid lines denote the pneumatic pressure lines connecting the supply and

Chetan S. Kulkarni et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

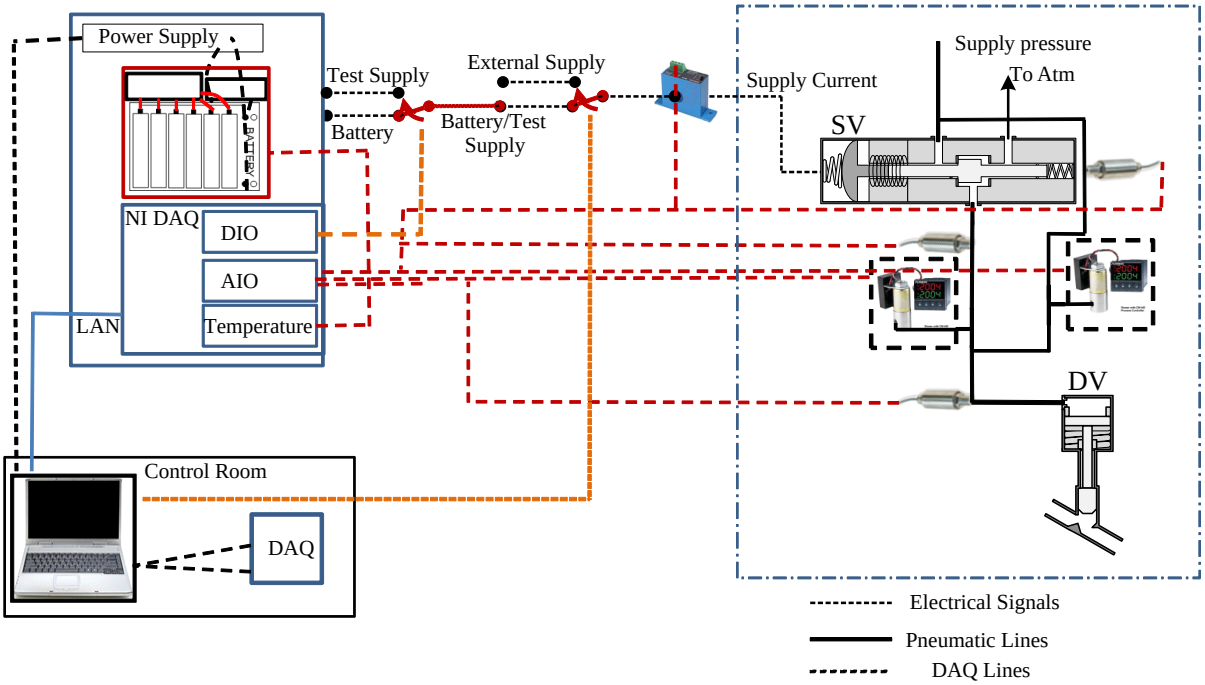


Figure 1. Prognostics demonstration testbed schematic.

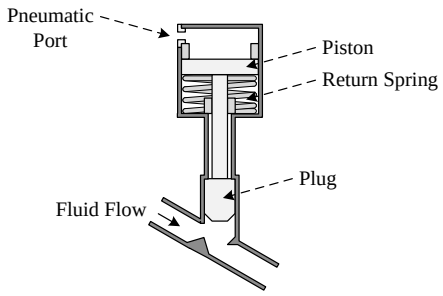


Figure 2. Discrete-controlled valve.

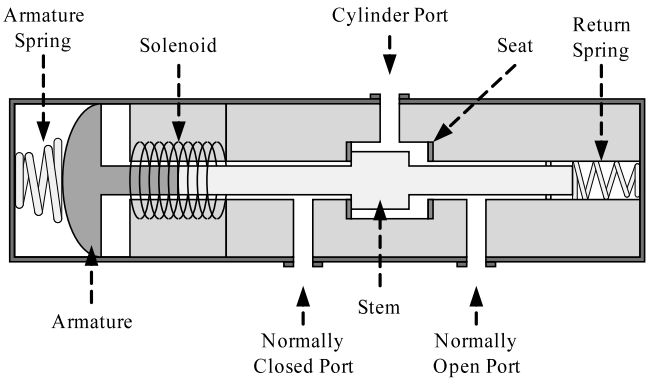


Figure 3. Three-way two-position solenoid valve.

the valves. Power is provided by both a typical power supply and a battery backup supply, and includes a fail-safe mode to isolate the valve prognostics testbed from the field cryogenic loading system interface.

The testbed includes a discrete-controlled valve (DV), illustrated in Fig. 2, which is a normally-open valve with a linear cylinder actuator. The valve is closed by filling the chamber above the piston with gas up to the supply pressure, and opened by evacuating the chamber to atmosphere, with the spring returning the valve to its default position.

A three-way two-position solenoid valve (SV), illustrated in Fig. 3, is used for controlling the operation of the DV valve. The cylinder port connects to the valve, the normally closed (NC) port connects to the supply pressure, and normally open (NO) port is left unconnected, allowing venting to atmosphere. When the solenoid is energized, the path from the NC port to cylinder port is open, allowing gas to pass from the supply to the valve, thus actuating the valve. When deenergized, the supply pressure is closed off and the path from the cylinder port to the NO port is opened, thus venting the actuation pressure in the DV valve, allowing the valve to open due to the return spring. The solenoid is powered by 24 V DC either through the power supply or the batteries.

The data from the different sensors is collected using an 8-slot NI cDAQ-9188 Gigabit Ethernet chassis as the data acquisition (DAQ) system that is designed for remote or distributed sensor measurements. For the testbed, control and data acquisition must be done remotely to meet safety requirements. A single NI CompactDAQ chassis can measure up to 256 channels of sensor signals, analog I/O (AIO), digital

I/O (DIO), and counter/timers with an Ethernet interface back to a host machine. All the operations for the cDAQ-9188 are controlled through an interface designed in LabVIEW. Additional details of the testbed and data aquisition system are described in (Kulkarni et al., 2013).

In this work, we focus on faults affecting the DV. Pneumatic valves can suffer from leaks, an increase in friction due to wear, and spring degradation (Daigle & Goebel, 2011a). Because friction and spring faults cannot be injected or their rate of progression controlled, we are limited only to leak faults, however, leaks are the most common faults found in pneumatic valves. In the configuration shown in Fig. 1, two different leak faults may be considered: (*i*) a leak to atmosphere, and (*ii*) a leak from the supply. In the former, this can manifest as a leak across the NO seat of the solenoid valve, or a leak in the pressure line going to the pneumatic valve. In the latter case, the fault can manifest as a leak across the NC seat of the solenoid valve. To emulate these faults, we installed two remotely-operated proportional valves, as shown in Fig. 1. One valve leaks to atmosphere (henceforth called the vent valve), while the other is installed on a bypass line around the solenoid valve (henceforth called the bypass valve).

The position of the vent and bypass valves can be controlled through a current signal, continuous between 0 and 100% open. In this way, we can control the fault progression (growth of leak size) according to various progression profiles.

Fig. 4 illustrates a leak to atmosphere using the vent valve (V1). The leak through V1 emulates a leak at the cylinder port or across the NO seat. Similarly, Fig. 5 illustrates a leak from the supply using the bypass valve (V2). The leak through V2 emulates a leak across the NC seat. The effect of these faults on valve behavior is described in Section 3.

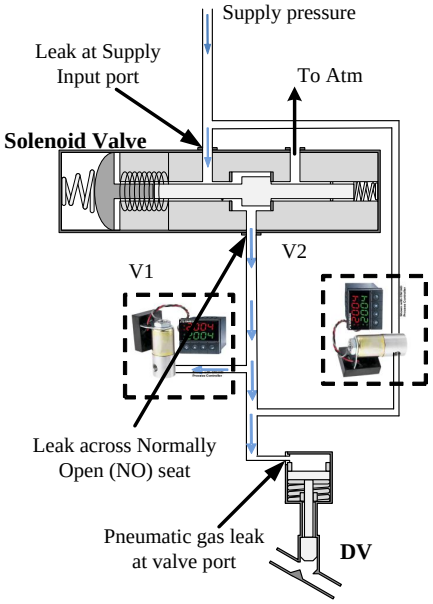


Figure 4. Solenoid valve leak fault injection when energized on DV valve.

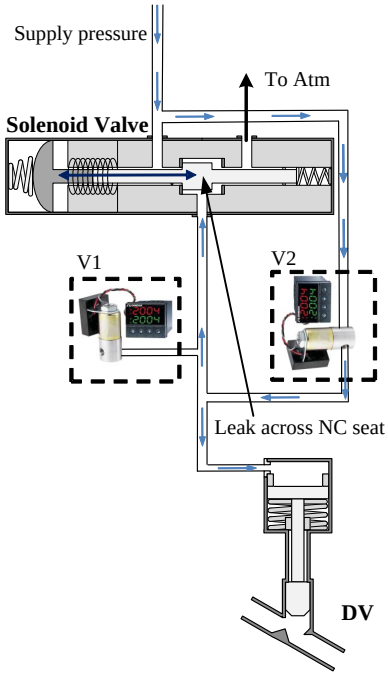


Figure 5. Solenoid valve leak fault injection when de-energized on DV valve.

3. VALVE MODELING

In the following section, we present the model using continuous-time. For implementation purposes, we convert to a discrete-time version using a sample time of 1×10^{-3} s. This model was originally presented in (Daigle et al., 2014), and we summarize it here for completeness.

We develop a physics model of the valve based on mass and energy balances. The system state includes the position of the valve, $x(t)$, the velocity of the valve, $v(t)$, the mass of the gas in the volume above the piston, and the mass of the gas in the pipe connecting the solenoid valve to the pneumatic valve port:

$$\mathbf{x}(t) = \begin{bmatrix} x(t) & v(t) & m_t(t) & m_p(t) \end{bmatrix}^T. \quad (1)$$

The position is defined as $x = 0$ when the valve is fully closed, and $x = L_s$ when fully open, where L_s is the stroke length of the valve.

The derivatives of the states are described by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} v(t) & a(t) & f_t(t) & f_p(t) \end{bmatrix}^T, \quad (2)$$

where $a(t)$ is the valve acceleration, $f_t(t)$ is the mass flow

going into the pneumatic port from the pipe, and $f_p(t)$ is the total mass flow into the pipe.

The single input is considered to be

$$\mathbf{u}(t) = \left[u_t(t)\right], \tag{3}$$

where $u_t(t)$ is input pressure to the pneumatic port, which alternates between the supply pressure and atmospheric pressure depending on the commanded valve position.

The acceleration is defined by the combined mass of the piston and plug, m, and the sum of forces acting on the valve, which includes the force from the pneumatic gas, $F_p = (p_t(t) - p_{atm})A_p$, where $p_t(t)$ is the gas pressures on the top of the piston, and A_p is the surface area of the piston; the weight of the moving parts of the valve, $F_w = -mg$, where g is the acceleration due to gravity; the spring force, $F_s = k(x(t) + x_o)$, where k is the spring constant and x_o is the amount of spring compression when the valve is open; friction, $F_f = -rv(t)$, where r is the coefficient of kinetic friction, and the contact forces $F_c(t)$ at the boundaries of the valve motion,

$$F_c(t) = \begin{cases} k_c(-x), & \text{if } x < 0, \\ 0, & \text{if } 0 \le x \le L_s, \\ -k_c(x - L_s), & \text{if } x > L_s, \end{cases} \tag{4}$$

where k_c is the (large) spring constant associated with the flexible seals. Overall, the acceleration term is defined by

$$a(t) = \frac{1}{m}(F_s - F_p - F_f - F_w + F_c). \tag{5}$$

The pressure $p_t(t)$ and the pipe pressure, $p_p(t)$, are calculated as:

$$p_t(t) = \frac{m_t(t)R_gT}{V_{t_0} + A_p(L_s - x(t))}p_p(t) \quad = \frac{m_p(t)R_gT}{V_p} \tag{6}$$

where we assume an isothermal process in which the (ideal) gas temperature is constant at T, R_g is the gas constant for the pneumatic gas, V_{t_0} is the minimum gas volume for the gas chamber above the piston, and V_p is the pipe volume.

The gas flows are given by:

$$f_{p,in}(t) = f_g(u_t(t), p_p(t)) \tag{7}$$
$$f_{p,leak}(t) = f_g(p_p(t), p_{leak}) \tag{8}$$
$$f_{p,t}(t) = f_g(p_p(t), p_t(t)) \tag{9}$$
$$f_p(t) = f_{p,in}(t) - f_{p,t}(t) - f_{p,leak}(t) \tag{10}$$
$$f_t(t) = f_{p,t}(t) \tag{11}$$

where $f_{p,in}$ is the flow into the pipe from the supply or atmosphere, $f_{p,leak}$ is a leak term with p_{leak} being the pressure outside the leak, $f_{p,t}$ is the flow from the pipe to the chamber above the piston, and f_g defines gas flow through an orifice for choked and non-choked flow conditions (Perry & Green, 2007). Non-choked flow for $p_1 \ge p_2$ is given by $f_{g,nc}(p_1, p_2) =$

$$C_s A_s p_1 \sqrt{\frac{\gamma}{ZR_gT}\left(\frac{2}{\gamma - 1}\right)\left(\left(\frac{p_2}{p_1}\right)^{\frac{2}{\gamma}} - \left(\frac{p_2}{p_1}\right)^{\frac{\gamma+1}{\gamma}}\right)}, \tag{12}$$

where γ is the ratio of specific heats, Z is the gas compressibility factor, C_s is the flow coefficient, and A_s is the orifice area. Choked flow for $p_1 \ge p_2$ is given by

$$f_{g,c}(p_1, p_2) = C_s A_s p_1 \sqrt{\frac{\gamma}{ZR_gT}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{\gamma-1}}}. \tag{13}$$

Choked flow occurs when the upstream to downstream pressure ratio exceeds $\left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)}$. The overall gas flow equation is then given by

$$f_g(p_1, p_2) = \begin{cases} f_{g,nc}(p_1, p_2) & \text{if } p_1 \ge p_2 \\ & \text{and } \frac{p_1}{p_2} < \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ f_{g,c}(p_1, p_2) & \text{if } p_1 \ge p_2 \\ & \text{and } \frac{p_1}{p_2} \ge \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ -f_{g,nc}(p_2, p_1) & \text{if } p_2 > p_1 \\ & \text{and } \frac{p_2}{p_1} < \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ -f_{g,c}(p_2, p_1) & \text{if } p_2 > p_1 \\ & \text{and } \frac{p_2}{p_1} \ge \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \end{cases} \tag{14}$$

The only available measurement is the valve position, so we have

$$\mathbf{y}(t) = \left[x(t)\right]. \tag{15}$$

Fig. 6 shows an example nominal valve cycle. The valve starts in its default open state. The valve is commanded to close at 0 s. Supply pressure (75 psig) is delivered to the pipe and to the valve, causing the piston to lower, closing the valve just after 1 s. At 4 s, the valve is commanded to open, and the pipe is opened to atmosphere. The pipe pressure and valve pressure drop, and once the pressure drops low enough, the spring overcomes the pressure force and the piston moves updwards. The valve completes opening just after 6 s. The valve parameters were identified from known valve specifications, and unknown parameters estimated to match the nominal opening and closing times, which for the actual valve, are both around 3.5 s.

As discussed in Section 2, we consider two different leak faults, one in which there is a leak from the supply pressure input to the valve (p_{leak} is the supply pressure), emulated us-

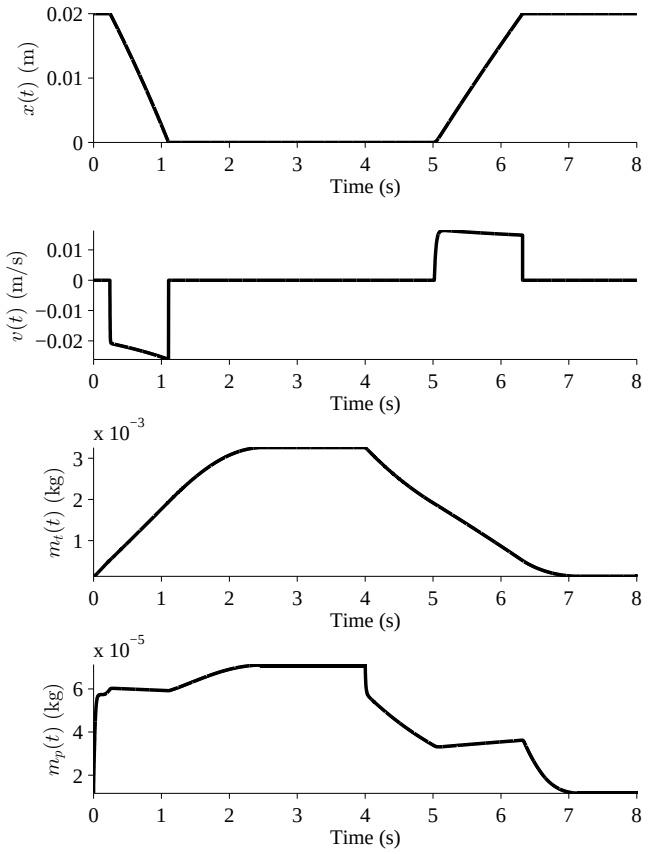


Figure 6. Nominal valve operation.

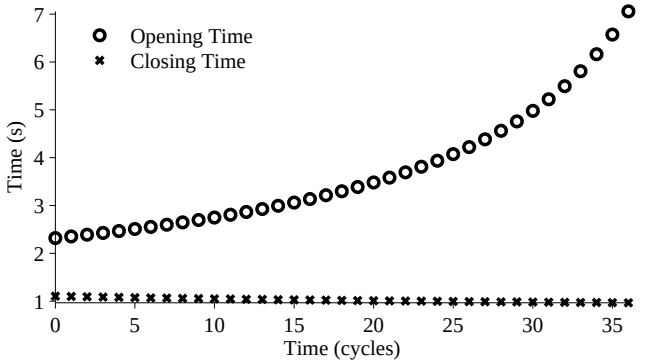


Figure 7. Valve timing with leak from supply, with linearly increasing leak area.

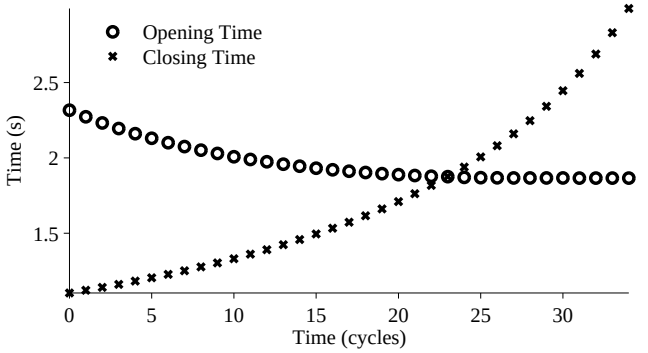


Figure 8. Valve timing with leak to atmosphere, with linearly increasing leak area.

ing the bypass valve, and one in which there is a leak out to atmosphere (p_{leak} is atmospheric pressure), emulated using the vent valve. In the former case, the valve will close more slowly and open faster, and in the latter, the valve will open more slowly and close faster. With a large enough leak, the valve may fail to open or close completely. Fig. 7 shows the changes in valve timing with the leak from the supply, and Fig. 8 shows the changes in valve timing with the leak to atmosphere. Here, we consider a damage progression model where the leak hole area increases linearly with time.

In the testbed, we cannot control the leak area, but only the leak valve position, which varies nonlinearly with the effective leak area. So, unlike in (Daigle et al., 2014), we must also consider this relationship, so that we can map from open/close times to leak size to leak valve position, for which we assume a particular damage progression profile. The relationship between the leak valve position and its effective area is a function of the valve flow coefficient, which is nonlinear. In this case, we assume that the effective area is equal to the product of the square of the position (A_{leak}^2) and a conversion coefficient.

$$kA_{leak}^2 = C_{leak} \tag{16}$$

We define valve end of life (EOL) through open/close time limits of the valves, as in real valve operations (Daigle & Goebel, 2011a). The valve in the testbed is required to open within 7 s and close within 6 s.

4. VALVE PROGNOSIS

We describe in this section the prognosis framework developed for the valve, following the general estimation-prediction framework of model-based prognostics (Luo, Pattipati, Qiao, & Chigusa, 2008; Orchard & Vachtsevanos, 2009; Daigle & Goebel, 2013). However, since we use only valve timing values for prognosis, we use a simpler estimation approach (Daigle et al., 2014), similar to that developed in (Teubert & Daigle, 2013), as opposed to more complex and computationally intensive filtering approaches used in previous works. We first formulate the prognostics problem, followed by a description of the estimation approach and a description of the prediction approach.

4.1. Problem Formulation

We assume the system model may be generally defined as

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \quad (17)$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \quad (18)$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[1]

In prognostics, we are interested in predicting the occurrence of some event E that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold $T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$ changes from the value 0 to 1 (Daigle & Sankararaman, 2013). That is, the time of the event k_E at some time of prediction k_P is defined as

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}. \quad (19)$$

The time remaining until that event, Δk_E, is defined as

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \quad (20)$$

In the context of systems health management, T_E is defined via a set of performance constraints that define what the acceptable states of the system are, based on $\mathbf{x}(k)$, $\boldsymbol{\theta}(k)$, and $\mathbf{u}(k)$ (Daigle & Goebel, 2013). In this context, k_E represents end of life (EOL), and Δk_E represents remaining useful life (RUL). For valves, timing requirements are provided that define the maximum allowable time a valve may take to open or close, and these define T_{EOL} (Daigle & Goebel, 2011a).

The prognostics problem is to compute estimates of EOL and/or RUL. To do this, we first perform an estimation step that computes estimates of $\mathbf{x}(k)$ and $\boldsymbol{\theta}(k)$, followed by a prediction step that computes EOL/RUL using these values as initial states. For the case of the valve, the future inputs are known, i.e., the valve is simply cycled open and closed, so there is no uncertainty with respect to future inputs.

4.2. Estimation

Since only valve position is measured, only valve timing values are useful for prognostics. We can obtain this information from the continuous position measurement data by extracting and computing the difference in time between when the valve is commanded to move, and when it reaches its final position. Using the model, we can map this time to the fault size that corresponds to it. In order to obtain this result quickly, we compute a lookup table that maps leak size to corresponding open and close times, by simulating the model given different leak sizes in the expected ranges. A similar approach is used for current-pressure transducers in (Teubert & Daigle, 2013).

We are interested in mapping this leak size back to the position of the leak valve, which we assume is increasing linearly. For this, we simply take the square root (Eq. 16). Since this transformed value is progressing linearly, we will essentially be estimating the gain term k, lumped with the slope of the leak valve position. So, given the estimated values of damage progression, we can perform a regression to find the line that fits this data, using the last N cycles.

For the leak to atmosphere, only closing times can be used (Daigle et al., 2014). This is because, in the presence of this leak, the valve may not get up to the full supply pressure when the valve closes in time for the next cycle, so since the internal valve actuator pressure is not measured, we do not have a correct initial condition for the simulation with which to estimate the leak parameter value for the following opening time. For the supply leak, we have analogous situation and can use only opening times for leak parameter estimation.

4.3. Prediction

Given the current estimated leak parameter value, and the regression parameters, we can compute the value of the leak parameter at any future time, defining the damage progression equation. Using the lookup table, we can map the maximum valve open/close times to maximum leak parameter values for the two leak faults, and this defines the EOL thresholds in the leak parameter space. Using the relationship between leak size and leak valve position, we can then obtain corresponding maximum values, and then solve for the time at which that threshold is crossed, given the fitted line, and thus obtain EOL.

Prediction is not performed until a fault is detected. To detect faults, we use a threshold on the opening times and closing times. If the mean valve opening or closing time, averaged over the last 3 cycles, is over the threshold, then a fault is detected. The regression is performed only over the data obtained since fault detection, so that nominal valve behavior is not used to estimate the fault progression parameters. The use of a filter on the data for fault detection introduces a slight lag, however in practice fault progression is very slow so this lag is negligible relative to the true EOL. In general, more robust fault detection strategies may also be used, but for our purposes a simple threshold works well.

We can isolate which fault is present by inspecting open/close timing trends (see Fig. 8 and Fig. 7). Since the two faults produce different qualitative changes on the valve timing, the observed trends tell us which fault is actually present.

[1] Bold typeface denotes vectors, and n_a denotes the length of a vector $\mathbf{a}$.

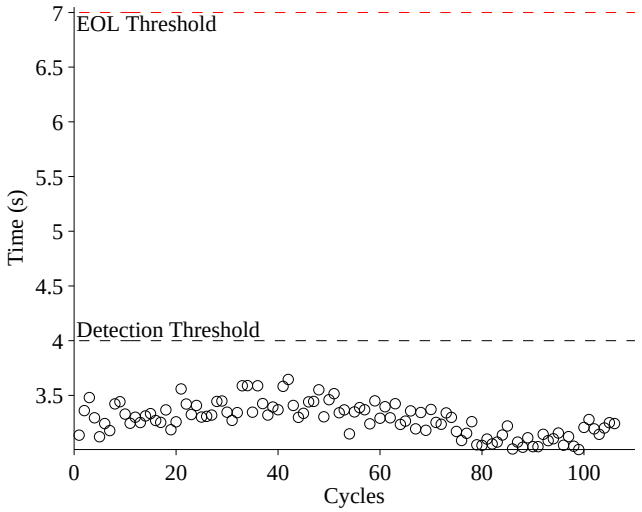


Figure 9. Valve open times with a atmoshpere leak.

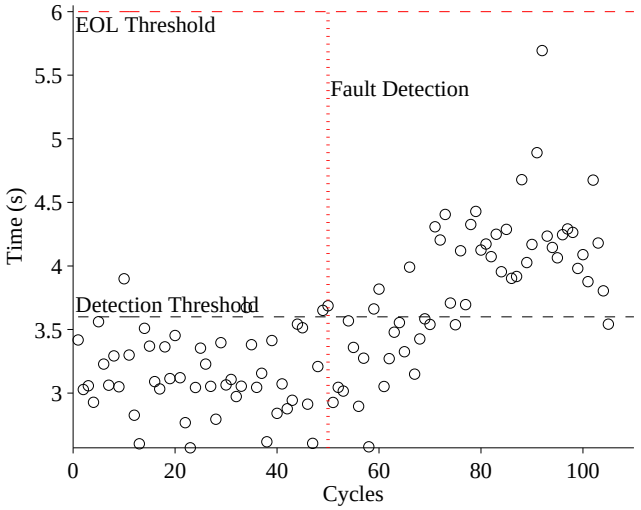


Figure 10. Valve close times with a atmoshpere leak.

5. RESULTS

We present here experimental results using the valve prognostics testbed. In each experiment, the valve is cycled open and closed repeatedly, every 10 s, until the end of life condition is reached. The valve under consideration is considered to be failed when it opens in 7 s or greater, or closes in 6 s or greater. Fault detection thresholds of 4 s and 3.6 s are used for the open and close times, respectively. The fault is injected by linearly increasing the open percentage of the desired leak valve in increments of 1%. We first present results for the leak to atmosphere fault, followed by results for the leak from supply fault.

5.1. Leak to Atmosphere

As described in Section 2, the leak to atmosphere fault is injected by controlling the position of the leak valve V1. This emulates a leak across the NO seat of the solenoid valve, or a leak on the gas line going to the pneumatic valve. As described in Section 3, this fault causes a decrease in opening times and an increase in closing times. Fig. 9 shows the open times of the valve during the fault progression, and Fig. 10 shows the close times. It is difficult to determine a trend in the open times, and they do not cross the detection threshold. The close times are very noisy, and do cross the closing time threshold at the 48th cycle. Based on the open and close times, the fault must be a leak to atmosphere, in agreement with the model.

The estimated leak parameter values, based on the close times of the DV, are shown in Fig. 11. In order to estimate the fault progression parameters, the last 50 values are used. Since the close times are quite noisy, a larger window is needed for this purpose. The RUL predictions are given in Fig. 12, where $\alpha = 0.3$ represents a desired accuracy constraint, and

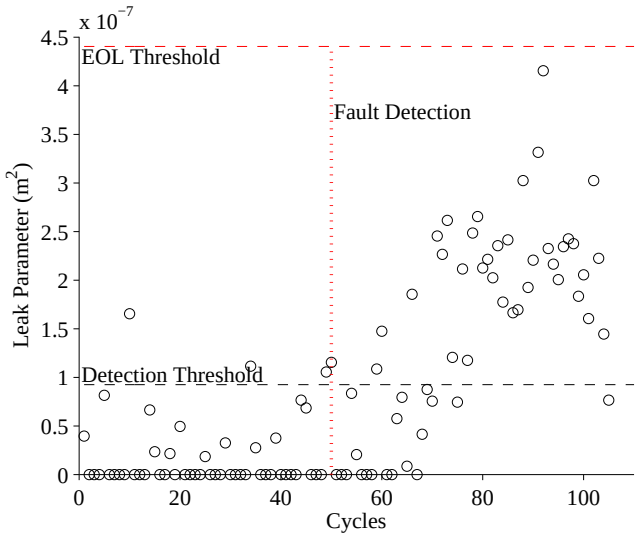


Figure 11. Estimated leak parameter values based on valve closing times for the atmospheric leak

RUL^* denotes the true RUL. The predictions converge relatively quickly after the fault is detected. The algorithm predicts RUL of the DV valve within the α-cone, until cycle 100. After that point, the close times have more spread, as can be seen from Fig. 10. Due to this, the algorithm overestimates the RUL values towards the end of the experiment.

5.2. Leak from Supply

As described in Section 2, the leak from supply fault is injected by controlling the position of the leak valve V2. This emulates a leak across the NC seat of the solenoid valve. As described in Section 3, this fault causes an increase in opening times and a slight decrease in closing times. Fig. 13 shows the open times of the valve during the fault progression, and

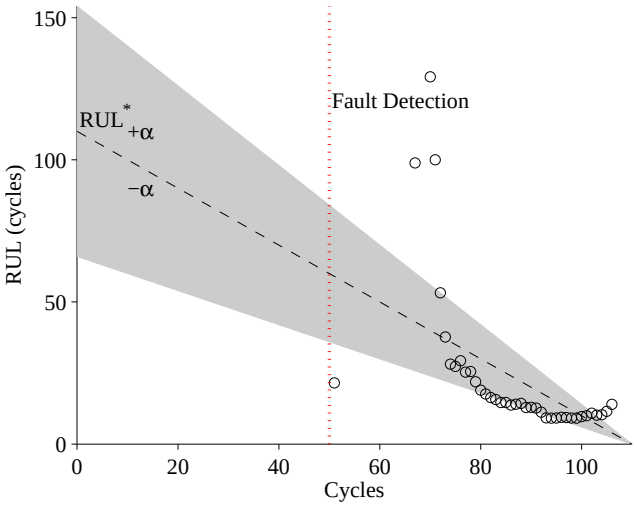


Figure 12. Predicted RUL values for the atmospheric leak.

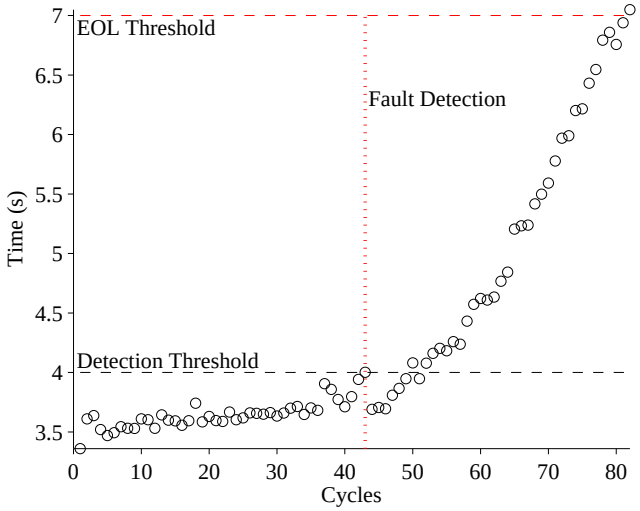


Figure 13. Valve open times with a leak from supply.

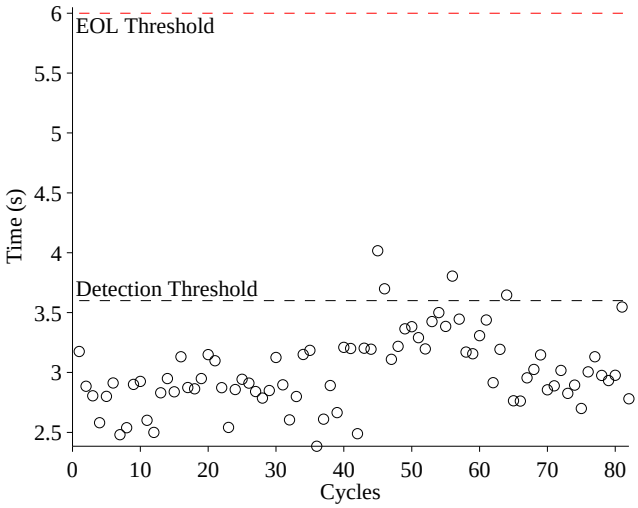


Figure 14. Valve close times with a leak from supply.

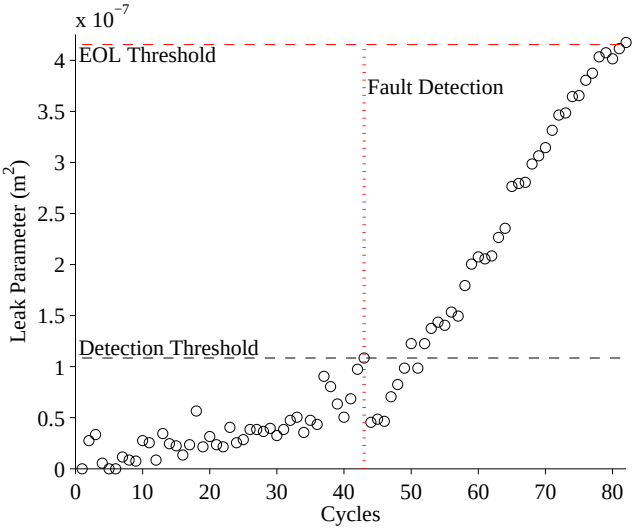


Figure 15. Estimated leak parameter values based on valve opening times for the leak from supply.

Fig. 14 shows the close times. The observed trends are in agreement with the model. A fault is detected at the 43rd cycle based on the opening times.

Fig. 15 shows the estimated leak parameters, and Fig. 16 shows the RUL predictions. After detecting the fault the predictions converge relatively quickly. Since the opening times are less noisy, only the past 15 cycles are used to determine the fault progression parameters, and this improves convergence. After entering the α-cone, the predictions for remain until EOL.

For further validation, we present a second experiment for a leak from the supply. The experiment is performed exactly the same, however, performance variations exist from one experiment to the next, and we must ensure that our approach is robust to those variations. The open and close times for this experiment are similar to the previous experiment, with some variations. In this case, the fault is detected later at around the 47th cycle in the opening times. The RUL predictions for this experiment are shown in Fig. 17. Although the valve timing is slightly different, the RUL predictions are just as accurate, and, in fact, a little more so in this case.

6. CONCLUSIONS

In this paper, we described a testbed for injecting faults in pneumatic valves. We developed a model of the valve including leak faults, and presented a valve prognosis framework that operates with limited measurements, using only valve timing information for prognosis. We demonstrated the prognosis framework with experimental data from the testbed for

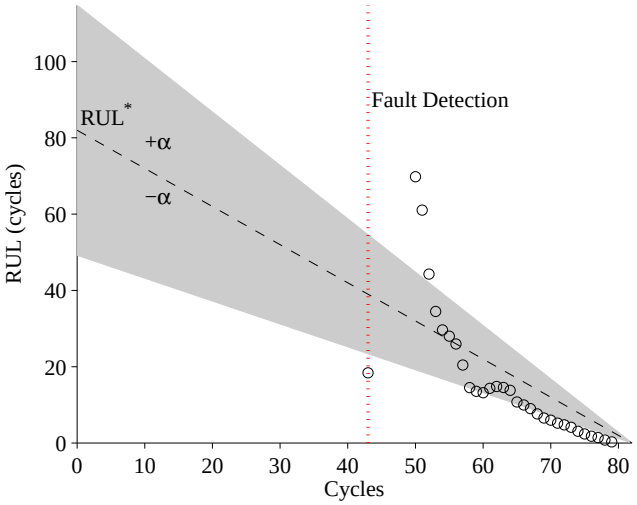


Figure 16. Predicted RUL values for the leak from supply.

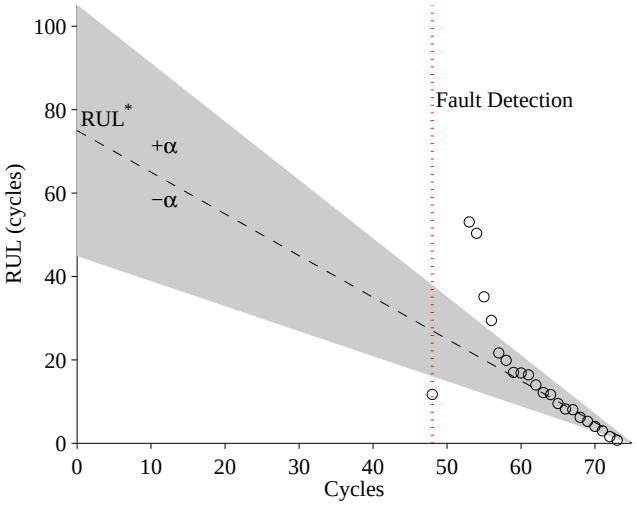


Figure 17. Predicted RUL values for the leak from supply (Exp. 2).

both types of leak faults, thus providing some validation of the approach.

Future work will involve validating the prognosis framework with additional experimental data from the testbed and applying the framework to faults occuring in continuously controlled valves.

ACKNOWLEDGMENT

This work was funded in part by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES), and by the Advanced Ground Systems Maintenance (AGSM) Project under the Ground Systems Development and Operations program.

REFERENCES

Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., & Gorospe, G. (2013). Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *International Journal of Prognostics and Health Management*, *4*(1).

Balaban, E., Saxena, A., Narasimhan, S., Roychoudhury, I., Goebel, K., & Koopmans, M. (2010, September). Airborne electro-mechanical actuator test stand for development of prognostic health management systems. In *Annual conference of the prognostics and health management society 2010.* Portland, OR.

Daigle, M., & Goebel, K. (2010, March). Model-based prognostics under limited sensing. In *2010 IEEE Aerospace Conference.*

Daigle, M., & Goebel, K. (2011a, August). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, *2*(2).

Daigle, M., & Goebel, K. (2011b, March). Prognostics for ground support systems: Case study on pneumatic valves. In *Proceedings of AIAA infotech@aerospace 2011 conference.*

Daigle, M., & Goebel, K. (2013, May). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Kulkarni, C., & Gorospe, G. (2014, March). Application of model-based prognostics to a pneumatic valves testbed. In *Proceedings of the 2014 IEEE aerospace conference.*

Daigle, M., & Sankararaman, S. (2013, October). Advanced methods for determining prediction uncertainty in model-based prognostics with application to planetary rovers. In *Annual conference of the prognostics and health management society 2013* (p. 262-274).

Kulkarni, C., Daigle, M., & Goebel, K. (2013, September). Implementation of prognostic methodologies to cryogenic propellant loading testbed. In *IEEE AUTOTESTCON 2013.*

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Orchard, M., & Vachtsevanos, G. (2009, June). A particle filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*(3-4), 221-246.

Perry, R., & Green, D. (2007). *Perry's chemical engineers' handbook*. McGraw-Hill Professional.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D.,

Lee, C., ... Koutsoukos, X. (2007, May). Advanced diagnostics and prognostics testbed. In *18th international workshop on principles of diagnosis* (pp. 178–185).

Poll, S., Patterson-Hine, A., Camisa, J., Nishikawa, D., Spirkovska, L., Garcia, D., ... Lutz, R. (2007, May). Evaluation, selection, and application of model-based diagnosis tools and approaches. In *AIAA infotech@aerospace 2007 conference and exhibit.*

Tang, L., Hettler, E., Zhang, B., & DeCastro, J. (2011). A testbed for real-time autonomous vehicle phm and contingency management applications. In *Annual conference of the prognostics and health management society 2011*.

Teubert, C., & Daigle, M. (2013, October). I/P transducer application of model-based wear detection and estimation using steady state conditions. In *Proceedings of the annual conference of the prognostics and health management society 2013* (p. 134-140).

BIOGRAPHIES

Chetan S. Kulkarni received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. He was a Senior Project Engineer with Honeywell Automation India Limited (HAIL) from 2003 till April 2006. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he has been a Staff Researcher with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and the IEEE.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

George Gorospe received the B.E. degree in Mechanical Engineering from the University of New Mexico, Albuquerque, New Mexico, USA, in 2012. Since October 2012, he has been a research engineer with the NASA Ames Research Center. His current research interests include mission design, systems engineering, and autonomous mobile robot control and control systems engineering.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively.

Dr. Goebel is currently the Technical Area Lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence and is the Technical Lead for Prognostics and Decision Making in NASA's System-wide Safety and Assurance Technologies Project. Prior to joining NASA in 2006, he was a Senior Research Scientist at General Electric Corporate Research and Development Center since 1997. He was also an Adjunct Professor of the Computer Science Department at Rensselaer Polytechnic Institute, Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems.

Dr. Goebel holds 15 patents and has co-authored more than 200 technical papers in the field of Systems Health Management. He is currently member of the board of directors of the Prognostics and Health Management Society and Associate Editor of the International Journal of Prognostics and Health Management.

Remaining Useful Life Prediction of Rolling Element Bearings Based On Health State Assessment

Zhiliang Liu[1], Ming J. Zuo[1,2], and Longlong Zhang[1]

[1] *School of Mechanical, Electronic, and Industrial Engineering, University of Electronic Science and Technology of China, Chengdu, P. R. China, 611731*

zhiliang_liu@uestc.edu.cn, loong125@yeah.net

[2] *Department of Mechanical Engineering, University of Alberta, Edmonton, Canada, T6G2G8*

mjzuo@ualberta.ca

ABSTRACT

Remaining useful life (RUL) prediction is one of key technologies to realize prognostics and health management that is being widely applied in many industrial systems to ensure high system availability over their life cycles. The present work proposes a data-driven method of RUL prediction based on multiple health state assessment for rolling element bearings. Instead of finding a unique RUL prediction model, the life cycle of bearings is clustered into three health states: the normal state, the degradation state, and the failure state. A local RUL prediction model is separately built in each health state. Support vector machine is the technology to implement both health state assessment (classification) and RUL prediction modeling (regression). Experimental results on two accelerated life tests of rolling element bearings demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

Bearings are the most common components in rotatory machines, and their failures are the most common failure cases in machinery. With increasing requirement of reliability, maintainability, testability, supportability and safety, extensive principles and models on the topic of bearing failure physics, diagnosis and prognostics have been reported in literature every year; however, most prognostic models do not have accuracy long-term prediction for the purpose of industrial applications, and thus prognostics techniques for remaining useful life (RUL) prediction are still quite challenging in both academia and industries (Kim et al. 2012, Siegel et al. 2011, Sun et al. 2011, Wang 2012).

Zhiliang Liu et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Ideally, RUL prediction can be viewed as a regression problem where a connection model between the sensitive features and the corresponding RUL is built over the complete life time. However, the methods using a unique regression model may be hard to represent the entire history and easily over fit the inconsistent patterns in some features (Wang 2012), because the trend of vibration based features is not necessarily monotonic with respect to degradation of bearings. In recent years, there is a trend that RUL prediction is suggested to be achieved individually on different health states (Kim et al. 2012). It implies the difference of intrinsic characteristics within different health states. Wang (Wang 2012) proposed two RUL prediction strategies to address the scenarios when the bearing faults have and have not been detected. Sutrisno et al (Sutrisno et al. 2012) realized degradation state recognition of bearings and estimated RUL based on making comparisons on durations of degradation states between the training and test bearings. Medjaher et al (Medjaher et al. 2012) proposed a data-driven method using mixture of Gaussian hidden Markova model (represented by dynamic Bayesian networks) to represent health states of bearings. Zhu et al (Zhu et al. 2013) proposed a performance degradation assessment method based on rough support vector data description. Siegel et al (Siegel et al. 2011) proposed a general methodology of how to perform rolling element bearing prognostics and presented the results using a robust regression curve fitting approach.

In the present work, we propose a RUL prediction method based on multiple health state assessment. Instead of looking for an overall regression model, we divide the entire bearing life into several health states where a local regression model can be trained separately. With the history life data from training bearings, we extract the characteristic features and knowledge about labels of health state, and then a classification model is built for health state assessment. We adopt SVM as the technique to implement

both health state assessment (classification) and local RUL prediction (regression), as SVM has been proved to be a suitable tool for both classification and regression problems.

2. PROPOSED METHOD

The proposed method includes two phases: training phase and testing phase. See Figure 1. The training phase generates a health state assessment model and local RUL prediction models corresponding to each health state. The testing phase uses the generated models from the training phase to estimate RUL when a new online sample is available.

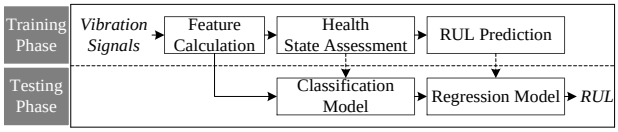


Figure 1. Flow chart of the proposed method

2.1 Health State Assessment

Health state assessment divides the whole life circle of bearings into several degradation states where RUL prediction model can be trained separately. In other words, the health state of a time point is recognized at first, and then the method adaptively selects the corresponding model to predict its RUL. The way of using health state assessment is the key idea of the proposed method, as we believe that RUL prediction models are not necessary the same in different health states. Instead of making great efforts to find a uniquely complex model for the whole life time, the piecewise approach based on health state assessment may be more practical.

In this section, we propose a hybrid approach that uses both unsupervised and supervised learning technologies to build a model for health state assessment. As no knowledge about health states is available at the very beginning of the data-driven method, we need to find a rough degradation states to supervise an accurate health state assessment. The idea is illustrated in Figure 2. We first use the unsupervised learning to extract knowledge about health state labels of all the time points. With the provided label knowledge, the supervised learning is employed to build a robust model of health state recognition.

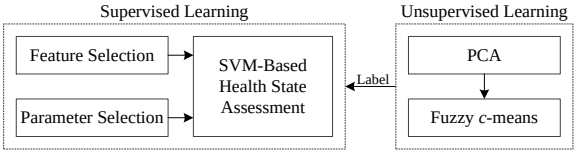


Figure 2. Hybrid approach for health state assessment

From the viewpoint of health state assessment, the run-to-failure data have their own intrinsic characteristics in different states. Therefore, we can use a clustering method to roughly group the run-to-failure data into L clusters, where L is the predefined number of health states and needs to be specified by users. Fuzzy c-means (Bezdek 1981), a classical method of clustering, is the suggested method of unsupervised learning in the proposed method.

Prior to fuzzy c-means, an unsupervised dimension reduction method is used to extract n' features from the original n features. In this method, principal component analysis (Shlens 2010), a well-known unsupervised technology of dimension reduction, is suggested to remove noisy features and reduce feature dimension while maintaining most of the variability from the original features (98 percentage of variability is used in this paper). By using the unsupervised learning, we can divide the bearing life into L health states by $(L-1)$ obtained thresholds, i.e. $t^1, t^2, \ldots, t^{L-1}$, as shown in Figure 3.

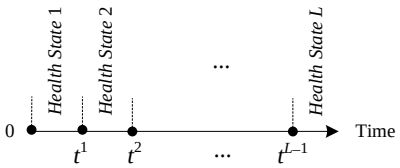


Figure 3. The proposed RUL prediction process

The proposed unsupervised approach can fuse many degradation features, and thus it usually provides a better performance than the approaches based on a single feature. In this paper, we specify the number of clusters to be three. The three health states, including normal state, degradation state, and failure state, are used to describe the bearing life duration. According to the time thresholds from unsupervised learning, we label the samples as one to represent the normal state if $t_i < t^1$, two to represent the degradation state if $t^1 \le t_i \le t^2$, and three to represent the failure state if $t_i > t^2$.

Then, the health state assessment becomes a supervised classification problem. In this paper, we use SVM as the classifier to build the model of health state assessment. Feature selection that aims to select an optimal set of features for SVM input can be implemented immediately after time record labeling. Parameter selection is also necessary to select the optimal parameters of SVM. Finally, the decision function of health state assessment is shown as follows:

$$\hat{STA} = sign\left(\sum_{i=1}^{p}\left[\alpha_i y_i \kappa(\mathbf{x}_i,\mathbf{x})\right] + \frac{1}{p}\sum_{i=1}^{p}\left[y_i - \alpha_i y_i \kappa(\mathbf{x}_i,\mathbf{x}_i)\right]\right), \quad (1)$$

where $sign(\,)$ is the sign function that extracts the positive or negative sign of a real number; κ is the kernel function; y is the label; α is the Lagrange multiplier; p is the number of support vectors. If $L \ge 3$, the health state assessment is a multiple class classification problem; therefore, the so-

called "one-against-all" approach (Vapnik 1995) is applied to the binary SVM in Eq. (1).

2.2 RUL Prediction

Based on the results from health state assessment, we train individual RUL prediction models on the degradation state and the failure state except the normal state. That is, we do not build the RUL prediction model for the normal state, as the normal state is quite diverse due to the different working condition. The RUL prediction is triggered only if the rolling bearings leave the normal state. By using the historical run-to-failure data, we can build RUL prediction models for the degradation state and the failure state. The technology to implement RUL prediction modeling is support vector machine that has also been used in (Sutrisno et al. 2012). Therefore, the RUL prediction value is computed as follows:

$$\hat{RUL} = \sum_{i=1}^{p}(\alpha_i - \alpha_i^*)\kappa(\mathbf{x}, \mathbf{x}_i) + \frac{1}{p}\sum_{i=1}^{p}\left[y_i - (\alpha_i - \alpha_i^*)\kappa(\mathbf{x}_i, \mathbf{x}_i) - \varepsilon \right], \quad (2)$$

where α and α^* is the Lagrange multiplier, ε is the margin of tolerance.

3. APPLICATIONS AND DISCUSSIONS

The proposed method is hereafter applied to experimental data that were collected from accelerated life tests (ALTs) of rolling element bearings. Those data have been used in the IEEE 2012 prognostic and health management (PHM) data challenge competition (Nectoux et al. 2012). The goal of the competition was to provide the best estimated RUL of rolling element bearings. One more thing to do before the following procedures is to clarify the failure criterion as it has great influence on the detailed modeling (Wang 2012). In the challenge, a bearing failure is deemed have happened if the amplitude of the vertical vibration signal exceeds a threshold of 20g (Nectoux et al. 2012).

Feature calculation is the following process after the signal preprocessing. As the failure criterion is vibration amplitude oriented, we define two related features that may reflect the degradation trend of rolling element bearings. The first one is the maximum absolute amplitude among the two vibration sensors. Taking the history vibration data into account, we define the second feature (called vibration-to-history index) as follows:

$$VH_i = \begin{cases} f_i, & if\ i = 1 \\ \dfrac{f_i}{\dfrac{1}{i-1}\sum_{j=1}^{i-1} f_j}, & if\ i > 1 \end{cases} \quad (2)$$

where f_i calculates the maximum absolute amplitude among the two sensors (the first defined feature); VH_i is the value of the vibration-to-history index on the ith time record.

Table 1 summarizes another 33 features adopted in this paper. Together with the specific two features, a total of 68 (33×2+2) feature values are extracted from the two vibration accelerometers. We then take the natural logarithm on all the 68 features to obtain possible linear trends, and another 68 new features are generated. Therefore, the total number of features used for the following process is 136. All the features are numbered from 1 to 136 sequentially. The first 66 features follow the same sequence in Table 1. The former half is from the horizontal accelerometer, and the latter half is from the vertical accelerometer. The 67th and 68th features are the two defined features, respectively. The last 68 features are organized the same as the first 68 features. The feature preprocessing including smoothing and normalization is conducted to continue process all the features. We use 11 as the fixed subset size in smoothing. Up to now, the features are ready for the use of both health state assessment and RUL prediction.

Table 1. Feature summary

Domain (#)	Feature
Time-domain (23)	• Fourteen conventional statistical features (Liu et al. 2013): maximum absolute value, average absolute value, peak to peak, root mean square (RMS), standard deviation, Skewness, kurtosis, variance, shape factor, crest factor, clearance factor, impulse factor, energy operator, and time series entropy; • Nine empirical mode decomposition (EMD) features (Dong 2012): RMS of the nine IMFs from EMD.
Frequency-domain (10)	• Six conventional statistical features (Liu et al. 2013): mean frequency, frequency center, rms frequency, standard deviation frequency, FFT entropy, and Hilbert entropy; • Four fault characterized frequency (Randall & Antoni 2011): ball pass frequency (outer race), ball pass frequency (inner race), fundamental train frequency (cage speed), and ball spin frequency.

In this application, the number of states is set to three. This choice is motivated by the fact that the degradation of the bearings can be represented by three health states: the normal state, the degradation state, and the failure state. With the extracted label knowledge, health state assessment turns to be a supervised classification problem, which is solved by support vector machine. It is worth pointing out that the unsupervised learning is for only the training phase, while the supervised learning is for both the training phase and the test phase. By the suggested feature selection algorithm (Liu et al. 2013), 11 features (i.e. the 71th, 79th, 44th, 69th, 73th, 11th, 72th, 112th, 91th, 24th, and 76th features) are selected for the SVM based health state assessment; 14 features (i.e. the 76th, 79th, 72th, 11th, 70th, 24th, 73th, 92th, 71th, 44th, 82th, 69th, 112th, and 86th features) are selected for the RUL prediction of the degradation state; and 4 features (i.e. the 126th, 90th, 58th, and 74th features) are selected for the RUL prediction

modeling of the failure state. In addition, parameters of SVM are all optimized by an analytical method (Liu et al. 2014) and grid search. We use two historical data, i.e. the bearing1_1 and the bearing1_2, to train the proposed method. This follows the same training and testing ways described in the IEEE 2012 PHM data challenge competition. In the next, we take the bearing1_3 as an example to introduce the rest process of the proposed method. Figure 4 shows the results of health state assessment for the bearing1_3. From Figure 4, the SVM based method of health state assessment performs well except the regions where a health state nearly changes. This phenomenon is caused by the randomness of the model in the transition regions between two health states. Farther away from the transition regions, the randomness becomes much less effective, and the model of health state assessment can work in a stable way.

Figure 5 shows the RUL prediction results by applying the proposed method to the dataset of the bearing1_3. In our strategy, no RUL prediction is made when the health state is estimated as the normal state. This explains that no values in a range from 0 second to about 11350 seconds are plotted in Figure 5. From Figure 5, RUL prediction in the range from 11350 seconds to 17320 seconds is not very match to the true RUL values. This could be possible, as the learning set was quite small while the life duration of all bearings was very wide (from 1 to 7 hours). Performing good estimates was thereby difficult and challenging. The efficacy of data-driven methods is highly dependent on the quantity and quality of system operational data (Kim et al. 2012). A significant amount of past knowledge of the assessed bearing is required because the corresponding failure modes must be known in advance and well-described in order to assess the current health state. However, there is only two bearing datasets for training in the challenge. Performance of the proposed method could be improved if more bearing training datasets are included.

In the next, we compare the proposed method with one reported methods following the same way defined in the challenge. No matter which technologies embraced in a method, the final objective to accurately predict RUL is the same pursue of all the methods. The challenge provides three measures to evaluate RUL prediction results from all the RUL prediction methods. Tables 2 summarize all the results of the two methods for RUL prediction. From the table, we can see that the proposed method performs better than Wang et al (Wang 2012) for the bearing1_3 while performs comparable for the bearing1_4.

Table 2. RUL prediction results

ID	Current Life (s)	True RUL (s)	Wang (Wang 2012)	The Proposed Method
Bearing1_3	18010	5730	490	**5842**
Bearing1_4	11380	339	10	**1109**

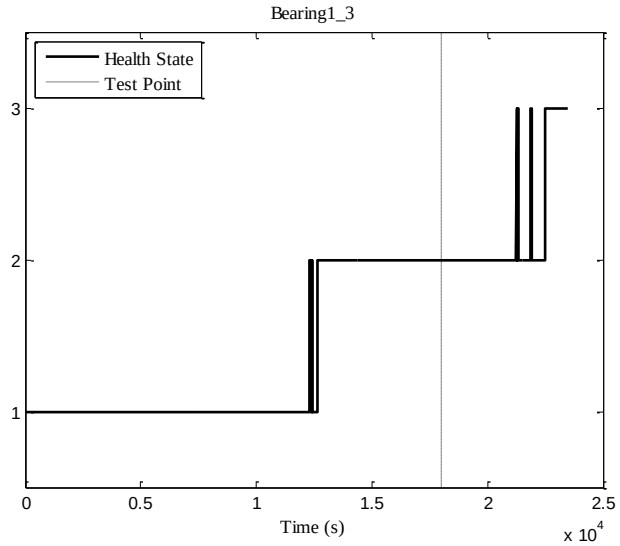


Figure 4. Health state assessment for the bearing1_3

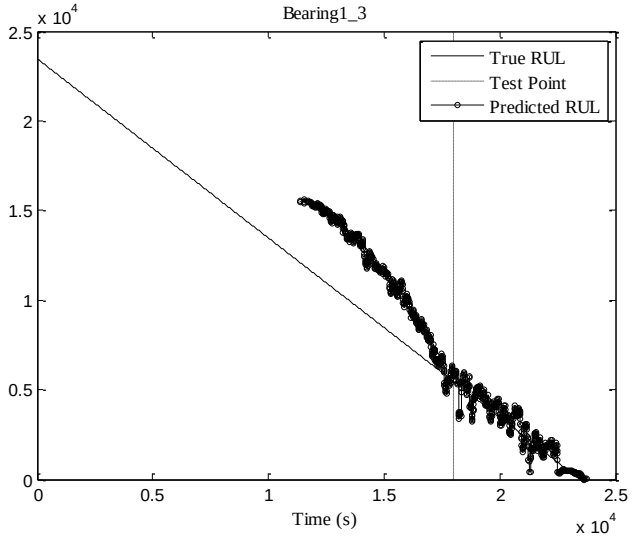


Figure 5. RUL prediction for the bearing1_3

4. CONCLUSIONS

In RUL prediction of bearings, the methods using a unique regression model may be hard to represent the entire history and easily over fit the inconsistent patterns in some features. Therefore, instead of looking for an overall regression model, this paper proposes a RUL prediction method based on multiple health state assessment. It basically includes four process steps: raw data collection, feature calculation, health state assessment, and RUL prediction modeling. With the help of health state assessment, the proposed method divides the entire bearing life into *L* health states where a local regression model can be built individually. As no knowledge about health states is available at the very beginning of the proposed data-driven method, we propose a

hybrid approach consisting of both unsupervised learning and supervised learning to estimate the health state of a bearing. The unsupervised learning with PCA and fuzzy *c*-means is used to automatically extract knowledge about health state labels of all the time points in the training phase. With the provided label knowledge, the supervised learning is employed to build a health state assessment model. SVM is the technology to implement both the supervised learning of health state assessment and RUL prediction modeling. Experimental results show the effectiveness of the proposed method.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (Grant No. 51375078), and supported by the State Key Laboratory of Rail Traffic Control and Safety (Contract No. RCS2014K006), Beijing Jiaotong University.

REFERENCES

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

Dong, S. (2012). *Research on space bearing life prediction method based on optimization support vector machine*, Doctoral dissertation. Chongqing University, Chongqing, pp. 1-129.

Kim, H., Tan, A. C., Mathew, J., Choi, B. (2012). *Bearing fault prognosis based on health state probability estimation*. Expert System Application. vol. 39, pp. 5200-5213.

Liu, Z., Zuo, M. J., Xu, H. (2013). *Fault diagnosis for planetary gearboxes using multi-criterion fusion feature selection framework*. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 227, pp. 2064 -2076.

Liu, Z., Qu, J., Zuo, M. J., Xu, H. (2013). *Fault level diagnosis for planetary gearboxes using hybrid kernel feature selection and kernel Fisher discriminant analysis*. The International Journal of Advanced Manufacturing Technology 67, pp. 1217-1230.

Liu, Z., Zuo, M. J., Zhao, X., Xu, H. (2014). *An analytical approach to fast parameter selection of Gaussian RBF kernel for support vector machine*. J Inf Sci Eng Accepted on Mar. 11, 2014.

Medjaher, K., Tobon-Mejia, D. A., Zerhouni, N. (2012). *Remaining useful life estimation of critical components with application to bearings*. Ieee T Reliability 61, pp. 292-302.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., Varnier, C., Others (2012). *PRONOSTIA: An experimental platform for bearings accelerated degradation tests*, IEEE International Conference on Prognostics and Health Management, Denver, CO, pp. 1-8.

Randall, R. B., Antoni, J. (2011). *Rolling element bearing diagnostics-a tutorial*. Mechanical System and Signal Processing 25, pp. 485-520.

Shlens, J. (2010). *A tutorial on principal component analysis*. Institute for Nonlinear Science, University of California, San Diego.

Siegel, D., Lee, J., Ly, C. (2011). *Methodology and framework for predicting rolling element helicopter bearing failure*, IEEE Conference on Prognostics and Health Management, Montreal, QC, pp. 1-9.

Sun, C., Zhang, Z., He, Z. (2011). *Research on bearing life prediction based on support vector machine and its application*. Journal of Physics: Conference Series 305, pp. 012-028.

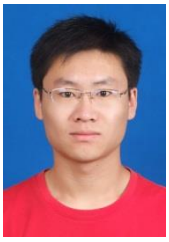
Sutrisno, E., Oh, H., Vasan, A. S. S., Pecht, M. (2012). *Estimation of remaining useful life of ball bearings using data driven methodologies*, Prognostics and Health Management (PHM), 2012 IEEE Conference on, Denver, CO, pp. 1-7.

Wang, T. (2012). *Bearing life prediction based on vibration signals: A case study and lessons learned*, IEEE Conference on Prognostics and Health Management, Denver, CO, pp. 1-7

Zhu, X., Zhang, Y., Zhu, Y. (2013). *Bearing performance degradation assessment based on the rough support vector data description*. Mechanical System and Signal Processing 34, pp. 203-217.

Vapnik V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, London.

BIOGRAPHIES



Dr. Zhiliang Liu received the B.S. degree in school of electrical and information engineering, Southwest University for Nationalities, Chengdu, in 2006. He visited University of Alberta as a visiting scholar from 2009 to 2011. He got his Ph.D degree on 2013 from the school of automation engineering, University of Electronic Science and Technology of China, Chengdu. His research interests mainly include pattern recognition, data mining, fault diagnosis and prognosis.

Dr. Ming J Zuo received the Ph.D. degree in 1989 in Industrial Engineering from Iowa State University, Ames, Iowa, U.S.A. His research interests include system reliability analysis, maintenance modeling and optimization, signal processing, and fault diagnosis. He is Associate Editor of IEEE Transactions on Reliability, Department Editor of IIE Transactions (2005-2008, 2011-present), Regional Editor for North and South American region for International Journal of Strategic Engineering Asset Management, and Editorial Board Member of Reliability Engineering and System Safety, Journal of Traffic and

Transportation Engineering, International Journal of Quality, Reliability and Safety Engineering, and International Journal of Performability Engineering. He is Fellow of the Institute of Industrial Engineers (IIE), Fellow of the Engineering Institute of Canada (EIC), Founding Fellow of the International Society of Engineering Asset Management (ISEAM), and Senior Member of IEEE.

Longlong Zhang received the B.S. degree and the master degree in School of Mechanical, Electronic, and Industrial Engineering, University of Electronic Science and Technology of China, Chengdu, in 2011 and 2014, respectively.

Data-Driven Lead-Acid Battery Prognostics Using Random Survival Forests

Erik Frisk[1], Mattias Krysander[2], and Emil Larsson[3]

[1,2,3] *Department of Electrical Engineering, Linköping University, Sweden*
frisk@isy.liu.se
matkr@isy.liu.se
lime@isy.liu.se

ABSTRACT

Problems with starter batteries in heavy-duty trucks can cause costly unplanned stops along the road. Frequent battery changes can increase availability but is expensive and sometimes not necessary since battery degradation is highly dependent on the particular vehicle usage and ambient conditions. The main contribution of this work is a case-study where prognostic information on remaining useful life of lead-acid batteries in individual Scania heavy-duty trucks is computed. A data-driven approach using random survival forests is proposed where the prognostic algorithm has access to fleet management data including 291 variables from 33603 vehicles from 5 different European markets. The data is a mix of numerical values such as temperatures and pressures, together with histograms and categorical data such as battery mount point. Implementation aspects are discussed such as how to include histogram data and how to reduce the computational complexity by reducing the number of variables. Finally, battery lifetime predictions are computed and evaluated on recorded data from Scania's fleet-management system.

1. INTRODUCTION

To efficiently transport goods by heavy-duty trucks it is important that vehicles have a high degree of availability and in particular avoid becoming standing by the road unable to continue the transport mission. An unplanned stop by the road does not only cost due to the delay in delivery, but can also lead to damaged cargo.

One cause of unplanned stops is a failure in the electrical power system, and in particular the lead-acid starter battery. The main purpose of the battery is to power the starter motor to get the diesel engine running, but it is also used to, for example, power auxiliary units such as heating and kitchen equipment. High availability can be achieved by changing batteries frequently but such an approach is expensive both due to frequent visits to a workshop and also due to the cost of the batteries. In addition, as will be shown, battery degradation is highly dependent on the particular usage and ambient conditions.

The main contribution of this work is a case-study, with methodological development and analysis results, based on fleet-management data from heavy-duty truck manufacturer Scania. A non-parametric and data-driven prognostics approach is used to compute, on an individual vehicle basis, prognostic information on remaining useful life of the lead-acid batteries in the vehicle. This information is then used to make dynamic and vehicle individual maintenance plans. The proposed approach mainly uses existing techniques but also some methodological development is done, in particular for handling histogram information and data reduction. The approach can be classified as a reliability function based prognostic approach (Linxia & Köttig, 2014).

The outline of the paper is as follows. First, Sections 2 and 3 introduces the case study and illustrates the characteristics of the studied problem and what problems that need to be solved to obtain a feasible solution. Section 4 then discusses the key step in the approach, how to estimate battery degradation properties based on fleet management data. One characteristic of the dataset is that it contains histogram variables and how they are introduced in the approach is discussed in Section 5. The fleet management dataset is large and Section 6 discusses how to extract the most important parts of the data to be used with the approach discussed in Section 4. Finally, Section 7 discusses how the proposed approach can be used in a prognostics and condition based maintenance setting and then some conclusions in Section 8.

2. PROBLEM BACKGROUND

There exist a number of approaches in the literature to do prognostics. One common approach is to look for trends in

Erik Frisk et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

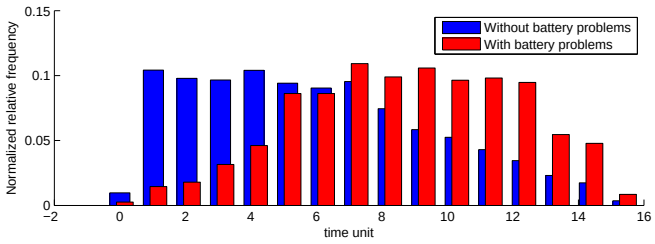


Figure 1. Normalized histogram of time stamp for vehicles with and without battery problems.

measured or estimated component health status indicators. Then, extrapolating computed health status indicators give indications on the amount of useful life left in the component. Such an approach requires reliable degradation models or measurements closely related to battery health, neither of which are available in this work. An alternative to a physics based approach where the battery health is estimated directly is to rely on recorded data from a large number of vehicles. This paper explores a data-driven approach where the prognostic algorithm has access to fleet management data and some characteristics of the data are

- 33603 vehicles logged from 5 different markets.
- 291 variables are logged for each vehicle.
- No time series, only aggregated data like traveled distance, year of delivery, histogram of ambient temperatures.
- Heterogeneous data; mix of numerical values such as temperatures and pressures with categorical data such as battery mount point or wheel configuration.
- Dataset includes histogram variables.
- Significant missing data rate ($\approx 15\%$).
- Each vehicle with a replaced battery has logged time of failure.
- There are many vehicles where battery failure has not occurred before the time of observation, i.e., data are right censored.

Figure 1 shows normalized relative frequency of logged time in the dataset. The red bars show the time of failure for vehicles with battery problems and the blue bars show time of logged data for vehicles with *no* battery problem. The histogram for vehicles with no battery problems thus reflect the last time data was logged from the vehicle which approximately is the age of the vehicle. Time is originally in days but has been scaled to *time units* to avoid revealing sensitive information. A first observation is that some batteries fail much earlier than others and that there clearly is potential in vehicle individual maintenance plans.

Let T be the random variable of failure time. Then the reliability function, sometimes referred to as the survival function, is the probability that $T \geq t$, i.e.,

$$R(t) = P(T \geq t) \tag{1}$$

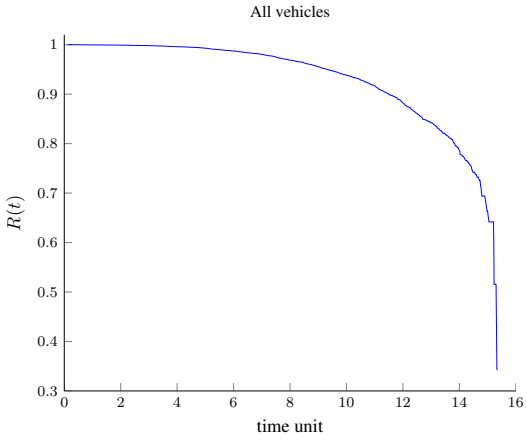


Figure 2. Reliability function estimate for the full dataset.

which is a fundamental object in the prognostics analysis. See Section 7 for further discussion on this. Estimating the reliability function from the data is basic survival data analysis and a non-parametric maximum-likelihood approach is used (Cox & Oakes, 1984). The reliability function estimate, based on the full dataset, is shown in Figure 2. This estimate would be most useful if it were true that the battery degradation is equal in all vehicles, no matter the vehicle configuration or usage. To investigate how much battery degradation characteristics change with vehicle configuration and usage, Figures 3 and 4 compare reliability function estimates for different subsets of vehicles. In Figures 3(a) and (b), different battery sizes and battery mounting positions are compared respectively. The reliability function estimate for battery size 140 Ah is based on very few vehicles, which is the reason for the jagged estimate. It is clear that battery size does not change the estimates significantly while battery mount position seems to have bigger impact. The battery size and battery position are both vehicle configuration parameters, naturally also usage parameters can have significant influence on battery degradation. Figure 4 shows reliability function estimates for vehicles with different amount of time with low battery voltage during cold ambient temperatures. Here it is clear that battery degradation significantly correlates with low temperatures and low voltages. The conclusion so far is then that truck battery degradation is dependent on vehicle usage and configuration. For each vehicle, 291 variables are recorded and it is not immediately clear which variables that are most important to describe different types of battery degradation profiles.

3. PROBLEM FORMULATION

The problem studied in this paper is to compute a probabilistic measure of the remaining useful life of a particular vehicle with a well functioning battery at a specified time $t = t_0$. As before, let T be the time of failure for the battery in a specific vehicle and let $\mathcal{V}$ denote usage and configuration data for the

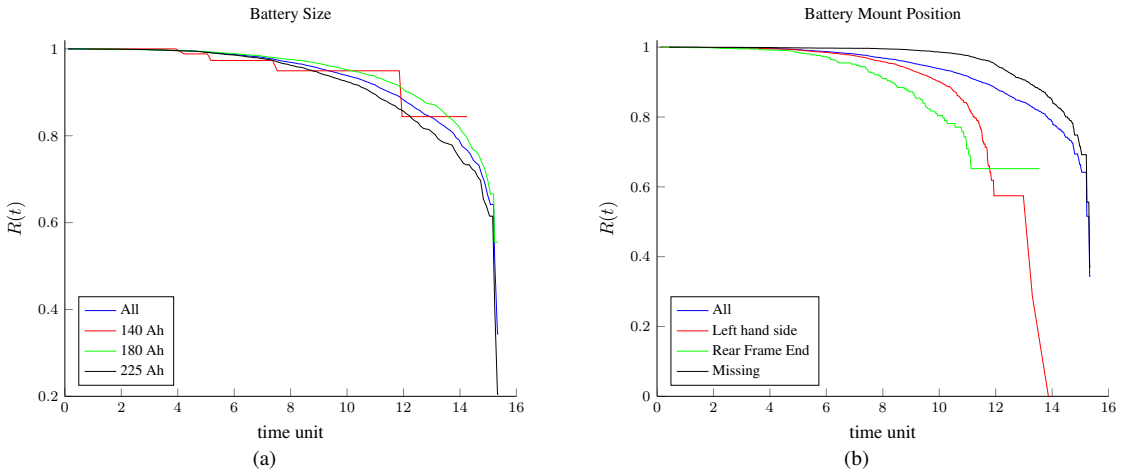


Figure 3. Reliability function estimation for different battery sizes (a) and different mounting positions (b).

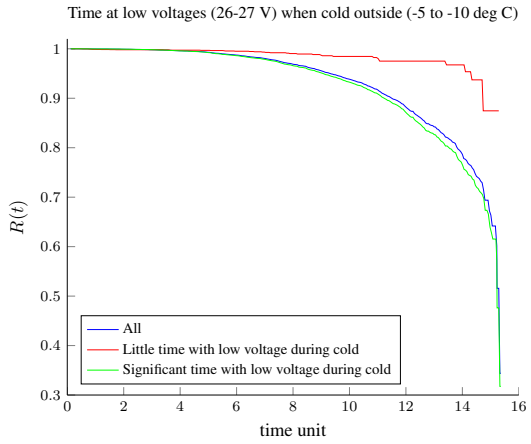


Figure 4. Reliability function estimate for vehicles with different amount of time with low battery voltage during cold ambient temperatures.

vehicle. The objective is to estimate the function

$$\mathcal{B}(t; t_0, \mathcal{V}) = P(T \geq t + t_0 | T \geq t_0, \mathcal{V}),\ t \geq 0 \qquad (2)$$

which describes, for a specific vehicle $\mathcal{V}$, the probability that the battery will at least t time units after t_0. This function is closely related to the reliability function $R(t)$. Let $R^{\mathcal{V}}(t)$ be the reliability function for a specific vehicle $\mathcal{V}$, then

$$\begin{aligned} \mathcal{B}(t; t_0, \mathcal{V}) &= P(T \geq t + t_0 | T \geq t_0, \mathcal{V}) = \\ &= \frac{P(T \geq t + t_0 | \mathcal{V})}{P(T \geq t_0 | \mathcal{V})} = \frac{R^{\mathcal{V}}(t + t_0)}{R^{\mathcal{V}}(t_0)} \end{aligned} \qquad (3)$$

The basic problem is then to, given the usage data for a vehicle $\mathcal{V}$, estimate $R^{\mathcal{V}}(t)$ and then compute $\mathcal{B}(t; t_0, \mathcal{V})$ according to (3). A key problem is that out of the 291 variables, it is not clear which ones that best capture different battery degradation characteristics. The main objectives of the paper are then to, in a case study with heavy-duty truck data,

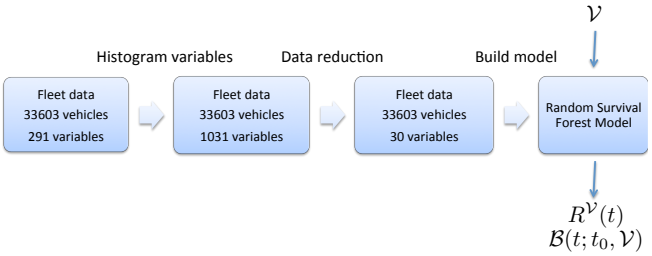


Figure 5. A flowchart describing the proposed approach.

- Determine, using machine-learning techniques, which of the 291 logged variables that are most useful for clustering vehicles with respect to battery lifetime prediction. Also analyze how to properly handle histogram variables.
- Estimate the reliability function $R^{\mathcal{V}}(t)$ for a specific vehicle $\mathcal{V}$.
- Estimate battery lifetime predictions as in (2) and evaluate on recorded data from Scania's fleet-management system.

The approach proposed for this problem is outlined in the flowchart in Figure 5. The flowchart illustrates how the original dataset first is extended with information about the histogram, which is described in Section 5. This leads to a significant growth in data size, which for complexity reasons results in a need to reduce the data before building models. The data reduction, here meaning selection of the 30 most important variables, is described in Section 6. Then, a random survival forest model is built as described in Section 4. With this model, a vehicle $\mathcal{V}$ and its associated 30 variables can be fed into the random survival forest model to compute prognostic information, which is illustrated in Section 7.

4. RELIABILITY FUNCTION ESTIMATION

Estimation of the reliability function (1) for a specific vehicle, based on a set of variables, is one of the main objective

of this work since then the function $\mathcal{B}(t; t_0, \mathcal{V})$ can be computed according to (3). As noted in Section 2, if it were a good assumption that battery degradation in all vehicles were independent on vehicle configuration, usage, and ambient conditions, a direct estimation of the reliability function using the very basic survival analysis techniques in (Cox & Oakes, 1984) would be appropriate. However this independence assumption is not realistic since it was shown how the failure rate of the battery varies significantly dependent on vehicle usage, configuration, and ambient conditions.

Thus, the 291 variables that are stored for each vehicle and describe vehicle configuration and usage need to be taken into account. One possibility is to use a parameterized approach where the failure rate of the batteries

$$h(t; \mathcal{V}) = P(T = t | T \geq t, \mathcal{V})$$

is written as a function of the variables $\mathcal{V}$. One common choice then is the proportional hazards model with log-linear hazards (Cox & Oakes, 1984) for which there exists well-established theory and tools. This approach is not used here, mainly because of the high rate of missing data which can not be handled directly, but also to avoid the proportional hazards assumption.

Instead, the basic idea of the approach used here can loosely be stated as utilizing a classifier to cluster vehicles with similar battery degradation properties. Then a non-parametric estimate for the reliability function $R^{\mathcal{V}}(t)$ is computed for a specific vehicle $\mathcal{V}$ using only the vehicles in the corresponding vehicle cluster.

A candidate tool that fits this situation well is Random Survival Forests (Ishwaran, Kogalur, Blackstone, & Lauer, 2008; Ishwaran & Kogalur, 2010). Random survival forest is a survival analysis extension of Random Forests (Breiman, 2001) which is a tree-based classifier (Breiman, Friedman, Stone, & Olshen, 1984) extended with bootstrap aggregation (Breiman, 1996) techniques. The key motives for using random survival forests in this work is that

- it handles heterogeneous data; both discrete and continuous valued variables
- it handles missing data
- it is non-parametric, i.e., does not rely on a specific hazard function parameterization like proportional hazards

There are 291 variables stored for each vehicle and the data includes 17 histograms. As will be described in Section 5, additional variables are derived to take these histogram variables into account. This results in a total of 1031 variables for each vehicle. To keep computational complexity down when building the random survival forest, Section 6 describes how to select the 30 most important variables. For this section it is not important exactly which variables that are used, it is enough to state that 30 variables were selected and used in the classifier.

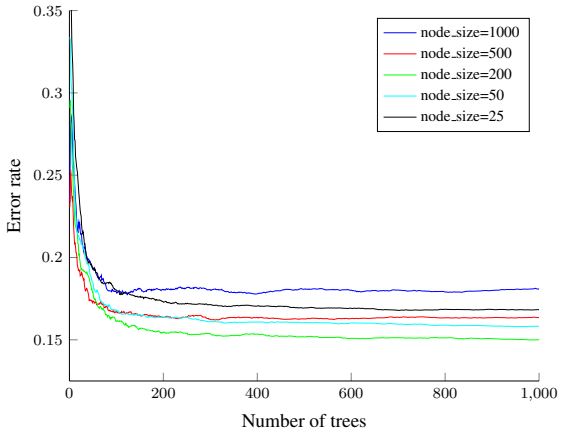


Figure 6. Error rate for the forest when node size is changed.

The experiments is conducted in R (R Core Team, 2014) using the package Random Forests for Survival, Regression and Classification (Ishwaran & Kogalur, 2013). There are 4 main parameters to be chosen in the software package

- number of trees to grow in the forest
- minimum size of terminal nodes
- number of random split variables
- number of random split values

Selection of these parameters is important for the result, and therefore there will be a short discussion on the choices made in this study. The remainder of this section requires knowledge of random survival forests, and for in-depth description of each parameter the reader is referred to (Ishwaran et al., 2008) and (Ishwaran & Kogalur, 2013).

The error rate measures how well the forest ranks two random individuals in terms of survival, and 0 is perfect and 0.5 is no better than guessing. The error rate can be interpreted as the probability of correctly ranking the survival of a batteries of two random vehicles. Formally, the error rate is $1 - C$ where C is Harrell's concordance index (Harrell, Califf, Pryor, Lee, & Rosati, 1982). Figure 6 plots the error rate as a function of node-size and number of trees. From this plot it is clear that, based on the error rate, there is no reason to grow more than about 200-300 trees in the forest and that the error rate is fairly insensitive to the selection of node size. The variance of the reliability function estimate dependends on the number of datapoints, i.e., too small terminal node sizes would give unreliable results. Based on Figure 6, the minimum terminal node size is chosen to 200.

The number of random variables to evaluate in each node of the tree classifier should not be too low, since then there is a lower probability of actually finding the best variable. Also, to get diversity among the trees in the grown forest, the number of variables should not be too high. As mentioned above and

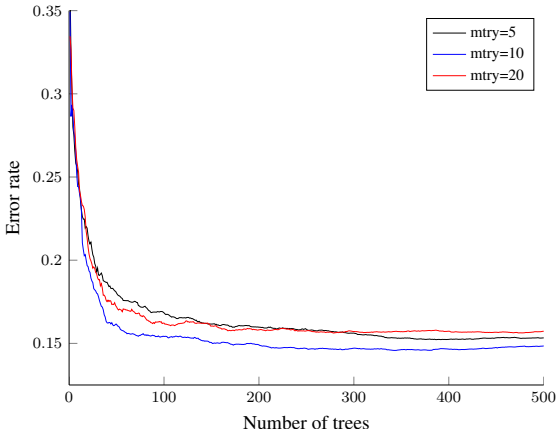


Figure 7. Error rate as a function of number of trees in the forest for three different number of random split variables to try in each node.

discussed further in Section 6, 30 variables are used in the analysis and Figure 7 shows the error rate for three different number of random variables explored in each tree node. Based on Figure 7, the number of random split variables to try in each node is selected to 10. The final parameter is the number of split values to try for each variable in each node. Due to the heterogeneous nature of the data, the package is configured for an exhaustive search for the best split value.

With the parameter values chosen, training the random survival forest with 200 trees, based on 30 variables for 33603 vehicles, takes about 15 minutes on the computer used for the experiments. The computer used has 128 GB of RAM and 2 Intel Xeon Processor X5675 (12M Cache, 3.06 GHz) resulting in 12 cores and 24 logical processors. In the experiment, 20 of the 24 logical processors were allocated in the tree computation. Note that training the forest is a one-time task, at least until more data becomes available, and predicting the reliability for a given vehicle is immediate.

5. HISTOGRAM VARIABLES

There are histograms in the available vehicle usage data and an example can be seen in Figure 10(a), which shows the fraction of time with a certain battery voltage. The frequencies of the observations in the intervals, the bin-values, are stored in the vehicle data. Thus, each bin-value is a variable that can be used for reliability function estimation.

By considering bin-values as independent variables, it is not taken into account that the bin-values represent frequencies of observations in intervals with known boundaries and that a histogram is an approximated probability distribution of a *single variable*. The mean and variance of a histogram are examples of properties that considers the underlying histogram variable and also take interval boundaries into account. Thus, could provide additional information for the reliability function estimation. To investigate properties of a histogram, a number of additional quantities, i.e., new variables, are derived for each histogram.

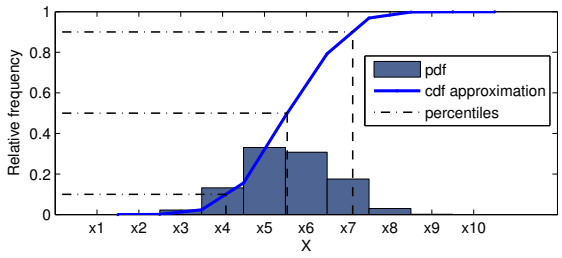


Figure 8. Histogram for a variable x.

Consider a histogram with n bins. Let p_i and x_i be the number of observations in and the center value of bin $i \in \{1, 2, \ldots, n\}$. The histograms are normalized such that the sum of bin values is one, i.e., $\sum_{i=1}^{n} p_i = 1$.

The variables considered for such a histogram are the bin values p_i for $i \in \{1, 2, \ldots, n\}$, the cumulative sum $c_i = \sum_{k=1}^{i} p_k$ for $i \in \{1, 2, \ldots, n\}$, the mean value of the histogram variable defined as $\mu = \sum_{i=1}^{n} p_i x_i$ and the variance

$$\sigma^2 = \sum_{i=1}^{n} p_i (x_i - \mu)^2$$

Furthermore the 10th, 50th (median) and 90th percentiles are computed from the cumulative distribution function based on a uniform distribution in each bin. Figure 8 illustrates the meaning of these values.

It is also natural that the tails, i.e., extreme cases of the distributions are of special importance. For example, a large number of starts with low battery voltage and almost none with high battery voltage could indicate battery problems. The following two variables have been included in the analysis to study the importance of the tails of the distribution.

Let the bin values of the mean histogram over all vehicles be denoted by $\bar{p}_i$ for $i \in \{1, 2, \ldots, n\}$. The number of bins that is considered as the left tail of the histogram n_- is computed from the mean histogram as $n_- = \max_n \sum_{i=1}^{n} \bar{p}_i < 0.05$. The number of bins considered as the right tail n_+ is computed analogously. Now, the tail variables considered for a histogram variable of a vehicle are computed as

$$\text{Ptail} = \sum_{i=1}^{n_-} p_i + \sum_{i=n-n_++1}^{n} p_i$$

and

$$\text{Mtail} = \sum_{i=1}^{n_-} p_i - \sum_{i=n-n_++1}^{n} p_i$$

6. VARIABLE IMPORTANCE

The dataset originally contains 291 variables where each bin in the histograms is counted as one variable. With the addition of the derived histogram variables described in Section 5 we obtain 1031 variables. To run the random survival forest algorithm considering the 291 variables takes 5 hours on the same machine that was described in Section 4. With 1031 variables, the computations did not finish in a reasonable time. To investigate parameter tuning of the forest, the algorithm has to be run with a number of different parameter settings. Then, also the run time with 291 variables is too long. To reduce computational complexity, the tree algorithms were run with 30 variables and this section describes how these variables have been selected.

To obtain accurate reliability functions it is important to use variables that are good at predicting battery failures. The predictive power of a variable will be called variable importance and this number can then be used to select the most important variables.

6.1. Method

Two different methods for computing variable importance have been investigated. The first method is based on the receiver operating characteristics curve, ROC-curve, and considers one variable at a time and the second is a multivariate analysis based on the error rate described in Section 4 computed by the random survival forest package.

Single variable analysis

The single variable analysis is based on the ROC-curve that shows the performance of a binary classifier. To introduce the ROC-curve, consider a hypothesis test concerning the battery of a vehicle with hypotheses

$$H_0 : \text{no battery problem}$$
$$H_1 : \text{battery problem}$$

For a variable x consider the test with threshold J and rejection region $\Phi(J) = \{x|x > J\}$ such that

$$\begin{aligned} x \notin \Phi(J) &: \text{accept } H_0 \\ x \in \Phi(J) &: \text{reject } H_0 \end{aligned} \tag{4}$$

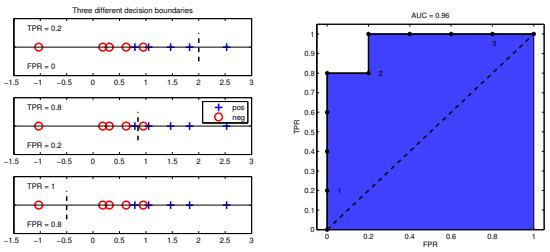
Two important properties of the test is the probability of detection, i.e.

$$P(D) = P(\text{reject } H_0 | H_1 \text{ is valid})$$

that ideally should be 1 and the probability of false alarm

$$P(FA) = P(\text{reject } H_0 | H_0 \text{ is valid})$$

which ideally is 0. Both the detection and false alarm probability is dependent on the threshold J and the ROC-curve is



(a) Example of variable values and 3 different thresholds.

(b) The ROC-curve for the dataset in Figure 9(a).

Figure 9. Example of an ROC-curve.

a plot of probability of detection $P(D)$ as a function of false alarm probability $P(FA)$. The curve is obtained by varying the threshold J.

An example of an ROC-curve is shown in Figure 9. Figure 9(a) shows the observations of a hypothetical variable used for classifying battery problems. The red circles are observations for vehicles without battery problems and the blue crosses observations from vehicles with battery problems. The value from vehicles with battery problems tends to be bigger than the values for vehicles without battery problem thus the variable could be used to separate those cases. The three different plots shows with a dashed vertical line different thresholds J and the true positive rates (TPR), i.e., the probability of detection, and the false positive rates (FPR), i.e., the probability of false alarm is shown.

The ROC-curve is shown in Figure 9(b) and is obtained by estimating the probabilities $P(D)$ and $P(FA)$ for thresholds J of different values. The numbers 1-3 refers to the 3 different thresholds shown in Figure 9(a). Consider for example the threshold in the second plot of Figure 9(a). Since 4 out of the 5 cases with battery problems are above this threshold the detection probability is estimation is $P(D) = 0.8$ and since 1 out of 5 cases without battery problems is above the threshold $P(FA) = 0.2$. This point is marked with a 2 in Figure 9(b). Variable importance for a variable x is then computed as the area under the ROC-curve (AUC) as

$$\text{AUC}(x) = \int_0^1 \text{ROC}(x)\, dx$$

For the example the AUC is 0.96.

The AUC is between 0 and 1. A value below 0.5 indicates that the observations from vehicles with battery problems are in general smaller than the observations of vehicles with fault free battery. In this situation a battery fault should be detected if the variable is below the threshold instead, i.e., to change the rejection region in equation (4) to $\Phi(J) = \{x|x < J\}$ and the AUC becomes 1 subtracted with the unmodified AUC. Hence all variables will get an AUC between 0.5 and 1 where a bigger value indicates a more important variable.

Multivariate analysis

Variable importance can also be computed using the error rate described in Section 4 as suggested in (Ishwaran et al., 2008, 2007). Variable importance for a specific variable x is evaluated by subtracting the error rate using all variables from the error rate obtained without using x. The error rate without x is evaluated on the original trees grown with x and whenever a split for variable x is encountered a daughter node is randomly assigned.

Advantages with this way of computing variable importance compared to the AUC-method is that the error rate is more closely related to our primary goal, i.e., to estimate the reliability function accurately and that the correlation of variables is considered. A disadvantage is the computational complexity of growing the trees needed to evaluate the error rates.

6.2. Case study results

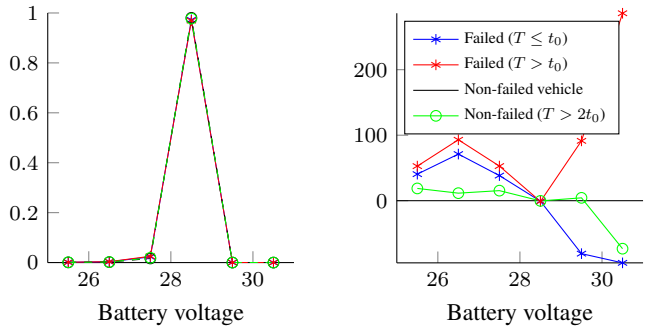
As said in the beginning of Section 6 the 30 most important of the total 1031 variables was selected as a trade-off between computational complexity and prediction performance. Since variable importance based on error rate requires the computation of a forest, the simpler AUC score has been used for the selection. The selection has been done in two steps. In the first step, the two most important variables of each histogram have been selected considering a variable correlation condition described later. In the second step, the 30 most important variables are selected among all non-histogram variables and the variables selected in first step. Since variable importance based on error rate is more closely related to reliability function prediction a comparison of the AUC-based ranking and error rate ranking is given in the end for of this section for the 30 selected variables.

Analysis of histogram variables

For each histogram stored in the dataset the variables described in Section 5 have been computed and the importance of them ranked according to the AUC.

Figure 10 shows an example of the mean histogram representing the relative time spent with a certain battery voltage when the battery temperature has been in the range of 10 to 25°C. To see how battery health effects the battery voltage the vehicles has been divided into 3 groups: vehicles with battery failure $T \leq t_0$, vehicles with battery failure $T > t_0$, and vehicles without any observed battery failure. Within the last set of vehicles also those with a long censoring time $T > 2\,t_0$ is shown separately. Figure 10(b) shows the relative deviation from the mean histogram under the fault free case. It can be seen that battery voltage is low more often for vehicles with battery failures.

Figure 11 shows variable importance based on AUC-score. The variables are introduced in Section 5 where pct stands



(a) Battery voltage histogram.

(b) Relative deviation in percent from fault free distribution.

Figure 10. Histogram for variable BattVoltTempI3, i.e., battery voltage when the battery temperature is in the range of 10 to 25°C.

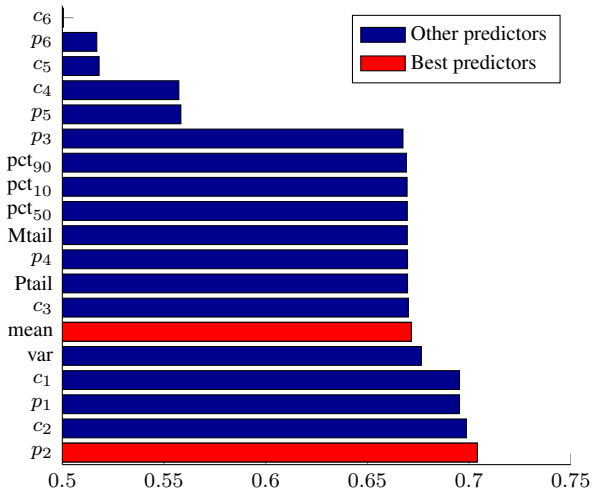
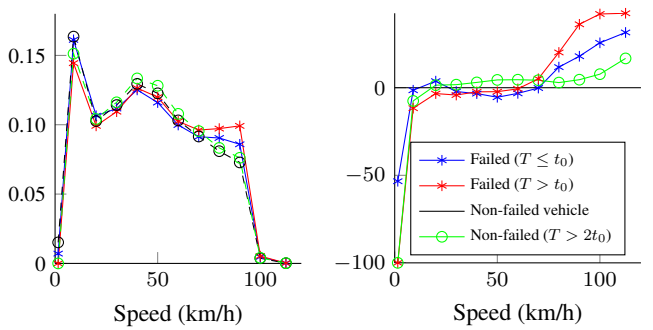


Figure 11. Importance of variables defined by the histogram for BattVoltTempI3 shown in Figure 10.

for percentile and Mtail and Ptail for minus and plus tail respectively. The most important variable of the histogram is p_2 which corresponds to the relative time with battery voltage between 26 and 27V. It can be seen that p_2 seems reasonable by looking at Figure 10(b) where the vehicle with failed batteries have a higher value than for the vehicles with non-failed batteries.

The next most important variable is c_2, i.e., the sum of the first two bins. Obviously c_2 is rather correlated with p_2 and to avoid the inclusion of highly correlated variables the most important variable is selected and the most important variable with a correlation with the most important variable less than 0.4. In this case, the mean value of the histogram will be the second selected variable.

For this histogram the original variable, p_2 was most important but the next histogram is an example where some of the derived variables are most important. Figure 12 shows a histogram for



(a) Histogram of speed in km/h when beginning to brake.

(b) Relative deviation in percent from fault free distribution.

Figure 12. Histogram for variable BrakeStartSpeed, i.e., initial vehicle speed when beginning to brake.

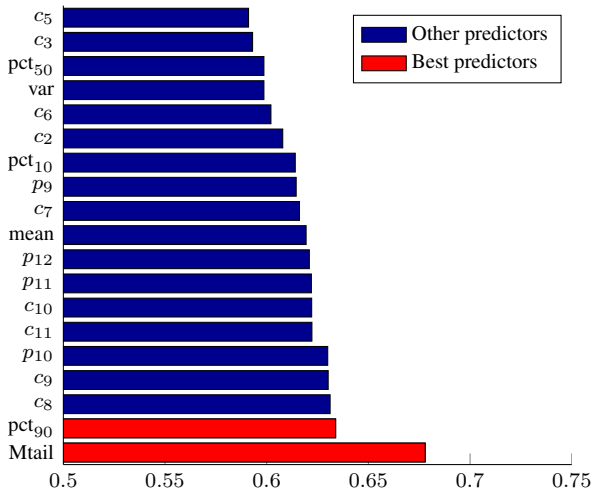


Figure 13. Importance of variables defined by the histogram for BrakeStartSpeed shown in Figure 12.

vehicle initial speed when beginning to brake. Figure 12(a) shows the histogram and Figure 12(b) the relative deviation from the mean histogram including only the vehicles without battery problems. Figure 13 shows variable importance for the variables related to the histogram in Figure 12. The most important variables here are the derived variables Mtail and pct90 and it can be seen in Figure 12 that vehicles with battery failures are more often beginning to brake at higher speeds.

As a summary of the histogram analysis, a number of variables has been derived for each histogram and two of the most important variables has been selected for each histogram when considering variable correlation. In the following analyses, only the two selected variables for each histogram will be considered together with all non-histogram variables.

Analysis of all variables

The remaining set of variables includes the selected histogram variables and the non-histogram variables and contains 117 variables. The 30 most important variables of these 117 variables are selected by using the AUC-based score and the top 18 are shown in Figure 14(a). The variables are categorized as bin variables p_i, non-histogram variables, or derived histogram variables. Among the selected 30 variables there are 5 non-histogram variables, 12 bin variables, and 13 derived histogram variables. Hence, some of the derived variables for the histograms are important. The individual variables with most predictive power are the total distance driven, time of delivery, and the number of days in use. The two most important bin variables are BattVoltTempI2_p2 which corresponds to low battery voltage at relatively low temperatures -5 to 10°C and BattVolt_p2 which corresponds to low battery voltage in general. The most important derived histogram variable concerns low ($< 20\%$) and high ($> 80\%$) state of charge when estimated after 8-24h without battery load. The variable importance based on error rate has also been computed of the top 30 variables in Figure 14(a) and the result is shown in Figure 14(b) where the top 18 variables are shown. Both rankings are quite similar. For example among the top 10 most important variables in each ranking 9 are the same. Thus even if the simpler AUC-based score has been used for variable selection the similarities with the more advance error rate based score is promising.

7. PROGNOSTICS AND CONDITION BASED MAINTENANCE

The main objective so far has been to compute the battery lifetime prediction function $\mathcal{B}(t; t_0, \mathcal{V})$ through estimation of the reliability function $R^{\mathcal{V}}(t)$ as described in Section 4 and then use (3).

With the reliability function and the battery lifetime prediction function, there are several ways to pass information to a condition based maintenance planner. One simple and direct way is to schedule the time for next maintenance T_{maint} no later than a time where the probability of a non-functioning battery is less than a certain threshold value. Formally,

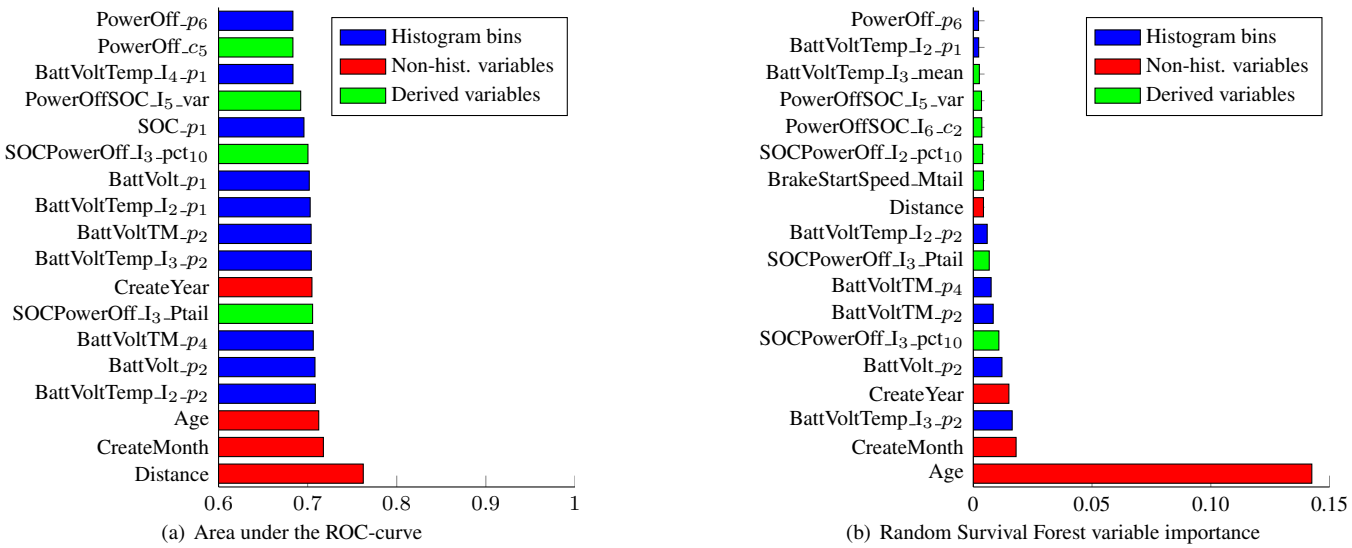
$$T_{\text{maint}} \leq \arg\min_t(\mathcal{B}(t; t_0, \mathcal{V}) < J) \tag{5}$$

where J is some predefined threshold. Another possibility is to compute the expected remaining useful life of the battery for a specified vehicle. Let $f(t)$ be the battery lifetime distribution. By definition it holds that $f(t) = -\frac{d}{dt}R(t)$ and then by partial integration

$$E(T) = \int_0^\infty t f(t)\,dt = -\int_0^\infty t\frac{d}{dt}R(t)\,dt = \int_0^\infty R(t)\,dt$$

This expression then gives that the expected remaining useful life of a battery in a vehicle $\mathcal{V}$, given that life up to $t = t_0$ is observed, is given by

$$E(\text{RUL}(t_0, \mathcal{V})) = \frac{1}{R^{\mathcal{V}}(t_0)}\int_{t_0}^\infty R^{\mathcal{V}}(t)\,dt - t_0$$



(a) Area under the ROC-curve

(b) Random Survival Forest variable importance

Figure 14. Individual predictive power for the most influential variables based on the area under the ROC-curve and ranking based on variable importance in the random survival forest.

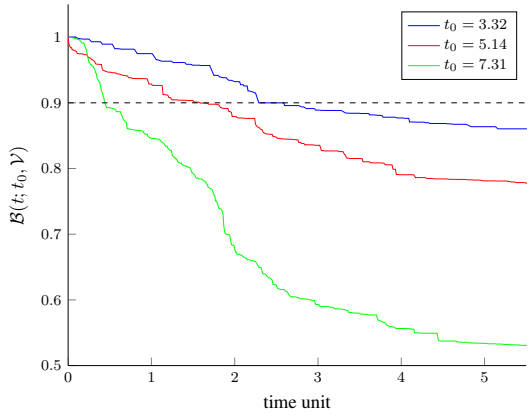


Figure 15. Function $\mathcal{B}(t; t_0, \mathcal{V})$ for three different vehicles with $t_0 = 3.32$, 5.14, and 7.31 time units respectively.

Although the expectation of remaining useful life is attractive, it involves integrating the estimated reliability function to infinity. Unfortunately, the estimated reliability functions has a high degree of uncertainty for large values of t. This is due to that there are very few recorded data points for large t and therefore this approach is not pursued further here. Instead, the battery lifetime prediction function is used as in (5).

Figure 15 shows the estimated $\mathcal{B}(t; t_0, \mathcal{V})$ function for three different vehicles selected from the set of all logged vehicles. For example, the figure shows how the probability of battery failure is increasing with increasing number of days in use. With a threshold of $J = 0.9$, the corresponding maintenance time T_{maint} should be no later than 2.29, 1.59, and 0.44 time units respectively. It is clear from Figure 15 that the expected battery lifetime prediction varies significantly for different vehicles. But that is to be expected since the three vehicles has been in operation significantly different amount of time.

In Figure 15 there are no confidence intervals or standard-error estimates. This is unfortunate since it is then difficult to assess how reliable the estimate of the reliability function is. To our knowledge, there is no standard way of estimating standard errors for bagged learners and random forests. Estimating confidence intervals for random survival forests is an active research area and one possible approach is described in (Wager, Hastie, & Efron, 2014).

To further investigate the impact on battery degradation from different usage profiles, ambient conditions, and vehicle configurations, Figure 16 shows the estimated battery lifetime prediction function for 20 vehicles with almost the same time in operation, about $t_0 = 5$ time units. Here it is clear that, even with similar time in operation the expected lifetime of the battery varies significantly. For example, comparing the vehicle with the worst predicted outcome with the vehicle with the best predicted outcome, the former vehicle has about 3% longer time in operation, which can not alone explain the big difference in predicted battery degradation. However, looking at the time with low battery voltages and low ambient temperatures, exactly as was done in Figure 4, it shows that the vehicle with worse battery lifetime prediction has spent significantly more time in that operating point. This also suggests that the dataset predicts that it is not sufficient to consider calendar time and mileage to get efficient vehicle individual maintenance plans.

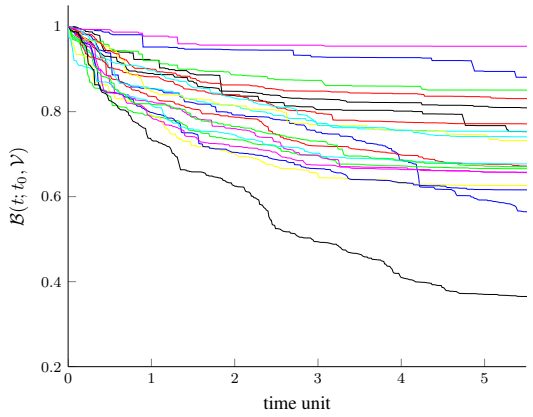


Figure 16. Battery lifetime prediction function $\mathcal{B}(t;t_0,\mathcal{V})$ for 20 vehicles with $t_0 \approx 5$ time units.

8. CONCLUSIONS

High degree of availability and reliability is important in many businesses and in particular heavy-duty trucks and the lead-acid battery is one important component to maintain. The battery is a difficult component to predict since degradation heavily relies on usage profile, vehicle configuration, and ambient conditions.

The main contribution is a case-study utilizing a data-driven approach to compute probabilistic reliability properties for a battery in a specific vehicle thus making condition-based maintenance feasible. The case-study is based on vehicle data from 33603 vehicles. A second contribution is the exploration of Random Survival Forests (RSF) for battery prognostics, and it is shown why RSF is a suitable tool in this application. A third main contribution is the study of which variables in the vehicle data that are important to characterize battery degradation. In particular a procedure is proposed how to include histogram data in the analysis.

The approach is evaluated using fleet-management data from truck manufacturer Scania and it is successfully shown how probabilistic reliability information can be estimated for the battery in individual trucks.

ACKNOWLEDGMENT

The authors acknowledge the initial work in the project done by Patrik Önnergren and Johanna Rosenvinge. This work was sponsored by Scania and VINNOVA (Swedish Governmental Agency for Innovation Systems) and the Swedish Research Council within The Linnaeus Center CADICS.

REFERENCES

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Cox, D. R., & Oakes, D. (1984). *Analysis of survival data* (Vol. 21). CRC Press.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, *247*(18), 2543–2546.

Ishwaran, H., & Kogalur, U. (2013). Random forests for survival, regression and classification (rf-src) [Computer software manual]. manual. Retrieved from `http://cran.r-project.org/web/packages/randomForestSRC/` (R package version 1.4)

Ishwaran, H., & Kogalur, U. B. (2010). Consistency of random survival forests. *Statistics & probability letters*, *80*(13), 1056–1064.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 841–860.

Ishwaran, H., et al. (2007). Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, *1*, 519–537.

Linxia, L., & Köttig, F. (2014, March). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, *63*(1), 191–207.

R Core Team. (2014). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org/`

Wager, S., Hastie, T., & Efron, B. (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, *15*, 1625-1651.

BIOGRAPHIES

Erik Frisk was born in Stockholm, Sweden in 1971. He received a PhD degree in 2001 from Linköpings University, Sweden. Currently he has a position as an associate professor at the Department of Electrical Engineering at Linköping University. His current research interests in the field of model based diagnosis, prognosis, and autonomous vehicles.

Mattias Krysander was born in Linköping, Sweden in 1977. He received a M.S. in electrical engineering in 2000, and a Ph.D. degree in 2006, both from Linköpings University, Sweden. Currently he has a position as an associate professor at the Department of Electrical Engineering at Linköping University. His current research interests in the field of model based diagnosis and prognosis. Fault isolation and sensor placement

Fatigue Crack Detection at Gearbox Spline Component using Acoustic Emission Method

Didem Ozevin[1], Justin Cox[2], William Hardman[2], Seth Kessler[3] and Alan Timmons[2]

[1]*Civil and Materials Engineering, University of Illinois at Chicago, Chicago IL*
dozevin@uic.edu

[2]*NAVAIR NAS Patuxent River MD*
alan.timmons@navy.mil
william.hardman@navy.mil
justin.cox@tsa.dhs.gov

[3]*Metis Design Corporation, Boston MA*
skessler@metisdesign.com

ABSTRACT

The spline section of helicopter gearbox structure is susceptible to fatigue crack, and non-redundant characteristic leads to the need for early flaw detection strategies. Acoustic Emission (AE) method relies on propagating elastic waves due to release of energy from active flaws. The initiation of damage is identified using the features of AE waveforms such as energy, amplitude and frequency centroid. The characteristics of the AE features are influenced by sensor type, sensor location and gearbox operational conditions. In this study, the AE data was collected from a helicopter gearbox with a notched spline section and realistic operational conditions using two different AE sensors located at two different positions. The data collection was conducted over one year under various operational conditions. The AE features were extracted from long duration waveforms (100 milliseconds) at every pre-defined time step (every 5 seconds). The frequency domain features of frequency centroid and energy distribution in various frequency bands were compared with gearbox operational conditions such as torque, lift, gyroscopic moment, and temperature. The influences of sensor location, sensor type and operational conditions on the AE features are presented in order to decouple their influences from the AE features due to damage. The comparison between the predicted crack growth time using the AE data and the observed crack initiation shows that the AE method using frequency domain features of streamed waveforms has great potential to identify the crack initiation when the sensor type and location are preserved.

Didem Ozevin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The gearbox components of the helicopters, especially the spline section, are prone to develop cracks and spalling due to excessive loads, insufficient lubricants, manufacturing defects, installation problems or material fatigue. It is important to design splines to prevent the onset of cracks, but inspection precautions such as early crack detection can prevent unexpected failures.

The common method to monitor flaws in splines is by visual inspection. Debris monitoring in an oil-wetted environment has had some success. Research indicates acoustic emission (e.g., Eftekharnejad and Mba 2011, Eftekharnejad et al. 2012, Li et al. 2012) and vibration signals (e.g., Yesilyurt et al. 2003) have better potential to detect spline damage if routine, automated inspections are performed. Acoustic emission inspections could relieve maintainers from the scrutinizing and subjective safety inspection requirements. Acoustic emission is based on propagating elastic waves released by active flaws. The sensors are typically mounted on the gearbox housing; therefore, propagating elastic waves pass through complex geometries, and interfaces of gearbox before reaching to the sensors. The method relies on searching for the presence of emissions due to damage as compared to operational noise emissions of gearbox, which are typically dominated by low frequency signals. The common sources that generate AE in gearbox include plastic deformation, microfracture, wear, bubbles, friction and impact (Li et al. 2012). For the vibration method, the progression of damage is extracted from time and frequency domain features of low frequency vibration data recorded by low frequency accelerometers in order to assess the changes

in vibrational properties as related to the damage (Li et al. 2002, Samuel and Pines 2005). The data processing can be enhanced further with multivariate pattern recognition methods (Wang 2008) and analytical understanding of gearmesh stiffness change with the tooth crack (Chaari et al. 2009, Chen and Shao 2011). Debris monitoring does not require any electronics, and is simple to interpret. The method has excellent sensitivity to wear-related failure, and in-line oil monitoring can detect spalling (Dempsey 2003); however, oil monitoring is insufficient to non-benign cracks as no debris is produced.

In order to increase the reliability of the measurement, two or more methods can be combined for redundant measurements. For instance Ozevin et al. (2006) implemented the combined acoustic emission/vibration sensors in the same package for concurrent data collection from gearbox components. In this study, waveforms are streamed at every selected time step instead of conventional threshold based approach with the idea of embedded high frequency crack emission into low frequency gearbox operational noise. Loutas et al. (2011) combined three methods as vibration, acoustic emission and oil-debris monitoring for rotating machinery. The authors applied principal component analysis (PCA) to reduce the number of parameters extracted from three methods, and concluded that the AE method is not sensitive to gear wear while the method detects the tooth crack earlier than the vibration method. Typical parameters extracted from the waveforms of AE and vibration are root mean square value, frequency domain characteristics, energy, spectral kurtosis, peak-to-peak vibration level, and ratio of the amplitude of the second tooth-meshing frequency. There are also advanced signal processing approaches such as wavelet decomposition of time domain data instead of traditional time domain features (e.g., Gu et al. 2011). However, the wavelet decomposition requires significant memory and slows the pattern recognition calculation if real time approach is implemented. Li and He (2012) developed empirical mode decomposition to the acoustic emission data for quantifying damage in gearbox. In majority of the studies in literature, the relations between damage and parameters are built based on the experimental data.

In this study, a comprehensive experimental design was conducted on an actual size gearbox and operational conditions. The AE data together with parametrics related to the operational conditions of the gearbox (e.g. temperature, forward load) were recorded over 130 hours. The two goals of this study are to (1) understand the influences of sensor type/location and gearbox operational conditions to the AE characteristics, (2) understand the relationships between the small and large crack sizes to the AE characteristics in comparison with the other measurements. It is important to determine and isolate the factors (e.g. gearbox temperature) influencing the AE features in order to develop the patterns in the AE data representing the crack growth only. The ultimate goal is to develop a repeatable real time pattern recognition approach to understand the condition of the gearbox spline component without recording waveforms but extracting and recording features from waveforms using field programmable gate array (FPGA).

2. EXPERIMENTAL DESIGN

In this section, the description of gearbox system and monitoring methodology are presented.

2.1. Description of Gearbox System

To replicate the failure progression with requisite complex loading and determine the required inspection intervals, NAVAIR-4.4.2 built the dedicated experimental test stand shown in Figure 1. Funding allowed three crack propagation tests to be performed to confirm that the test procedures produced representative fatigue surface topography. The three tests also provided a measure of statistical variability. In this paper, one test result was presented. The results obtained in this test were observed in other tests as well.

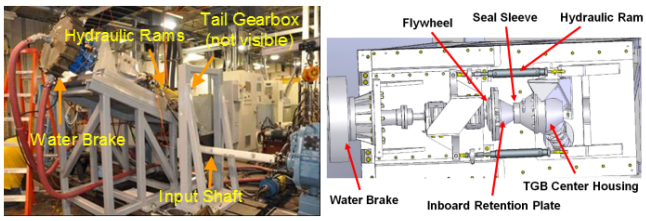


Figure 1. The experimental test stand.

The 2.5hr block cycle in the controlled environment simulated 2.5 flight hours (i.e., an average mission). The bench test included standard sensors for determining crack growth rates, finite element (FE) model calibrations, and development of a sensor system with algorithms for field inspections. These sensors were both internal and external to the gearbox. The sensors included strain and crack gauges, proximity probes, thermocouples, accelerometers, load sensors, and novel sensors such as energy harvesting, acoustic emission and guided wave sensors, thermal camera readings and pressure film for bolt preload. In addition, physical replicas of the spline surface tracked crack length and growth as the test progressed.

The bench test required machining a notch at the common field failure location in a spline to produce a stress riser. The current UT procedure for the spline easily detects this notch, which was made by electric discharge machining (EDM). Loading the test specimens independently on a 4-point bending test rig initiated a small subsurface crack from the notch feature before gearbox assembly. The full-scale test applied a flight-representative, multilevel block cycle with torque, thrust, and bending loads to the gearbox. The hub moment is the primary driver of the long crack growth rates, and it creates a one-per revolution cyclic stress like a

misaligned shaft. Because the hub moment can occur at any orientation, testing applied alternating force directions to evaluate the best and worst case sensor placements. These loads represented nonaccelerated, average mission loading.

2.2. Acoustic Emission Sensors and Monitoring Methodology

The AE system consists of PCI-2 data acquisition system, and two different sensor types including WD and micro-30 sensors. Both data acquisition system and sensors are manufactured by Mistras Group Inc. WD sensor has wideband response spanning 100 kHz to 1 MHz; micro 30 sensor has the bandwidth of 150 kH – 400 kHz. The AE sensors are coupled using vacuum grease and their locations are secured with aluminum brackets. Two sensors of each type are placed at different locations on the gearbox to understand the influence of sensor position relative to the radial load vectors on the bearings. Figure 2 shows the locations of the sensors around the periphery of the gearbox housing.

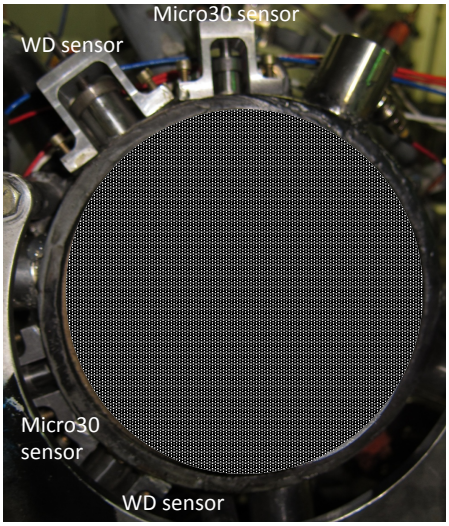


Figure 2. The sensor locations on the gearbox.

There are two approaches to collect the AE data: threshold-based and time-based, Figure 3. The threshold-based approach requires a pre-defined threshold level that the AE system acquires data when the signal level is above the pre-defined threshold. If threshold level is high, the sensitivity to detect micro-crack is reduced. If threshold level is low, the system may be overloaded by the data flow. The threshold-based approach has limitations for highly noisy applications where separating extraneous noise due to the operation of the system from relevant emissions generated by crack growth is a challenge. Time-based approach is independent from threshold. AE waveforms and features are recorded at every selected time interval. In this study, long duration (100 ms) waveforms are collected at every 5 seconds. The crack growth is a stochastic process. It is predicted that the crack emission will sum up with the operational noise and manifest itself in frequency domain features.

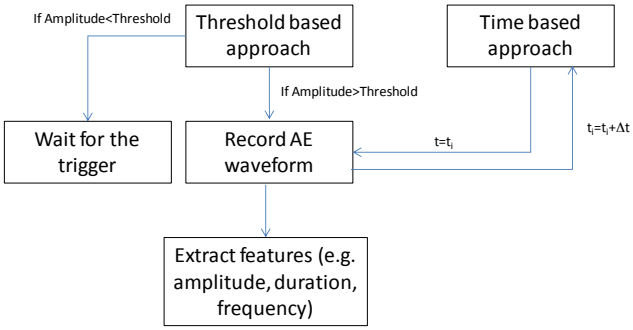


Figure 3. The comparison of threshold-based and time-based approaches.

The time-based waveform approach requires non-classical approach for damage detection. For example, cumulative hit or energy is not useful as each hit is recorded based on the pre-defined time interval. As the amplitude and other time domain features are influenced from operational noise, it is also difficult to extract the damage information using time domain features. In this study, patterns of frequency domain features are investigated in order to identify the variations in trends as indications of damage. The fundamental frequency domain features are frequency centroid and partial powers, Figure 4. The frequency centroid informs about the frequency content of a given waveform whether dominated by low frequencies or high frequencies. The partial powers are calculated by dividing the frequency spectrum into segments, and the area under each segment normalized to the total area represents partial powers. Frequency domain features allow monitoring the frequency contents of AE waveforms without recording them in real time, which requires extensive usage of memory, and is not feasible for real time pattern recognition approach.

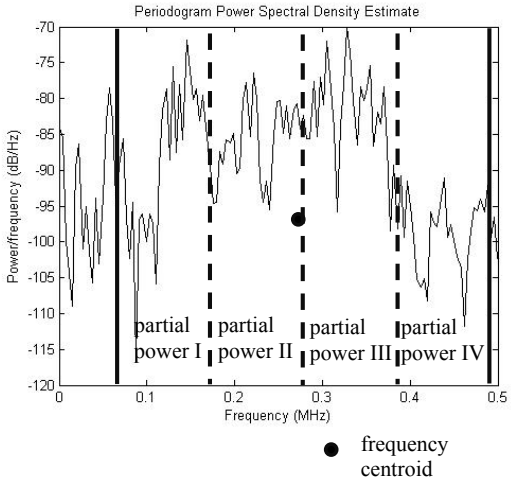


Figure 4. An example of frequency spectrum with frequency domain features.

3. ACOUSTIC EMISSION RESULTS

The acoustic emission data are first analyzed using individual waveforms, and time and frequency domain features are extracted from the waveforms in order to obtain the feature patterns throughout testing. The total duration of the analyzed data is about 130 hours of gearbox operation. The extracted features are compared with the gearbox operational parameters including temperature and hub moment.

3.1. Waveform Analysis

Figure 5 compares the frequency spectra of four sensors detected at different times of testing. The spectral energy of WD sensors is spread in the range of 20 kHz-500 kHz while the spectral energy of micro 30 sensor is dominated by frequencies lower than 400 kHz. As the sensors are resonant type sensors, their transfer function significantly modifies the output signal. Additionally, for the identical sensor types, there are slight differences in frequency spectra because of the influence of the sensor location. Therefore, the pattern recognition results presented in this study are limited by particular sensor type and location on the gearbox. This is the major limitation of selecting resonant type sensors in the experimental program.

A slight shift of the frequency spectrum to higher frequencies is observed for channels 1 and 3 when the test is progressed (crack was expected to grow by then). Those channels are placed next to each other. There is no significant change observed for channel 2. The mid-frequencies for channels 3 and 4 have the reduced energy for the day 21. The review of individual waveform requires significant amount of computational time. In next section, features are extracted from frequency domain features to understand the history of features in comparison with the gearbox operational conditions.

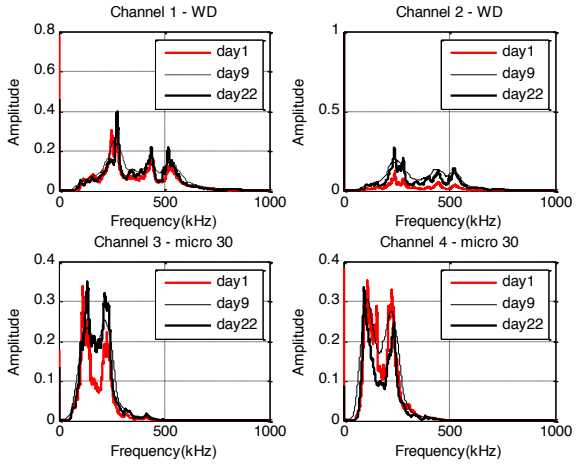


Figure 5. Frequency spectra of four sensors recorded at three different days of testing.

3.2. Feature Analysis

The AE amplitude histories of four sensors are shown in Figure 6. Throughout the monitoring period of over 120 hours, there is no significant chance in amplitudes observed. This shows that the AE amplitude is not a relevant feature to monitor the small crack growth. As discussed earlier, the AE amplitudes are controlled by operational conditions, which cause high amplitude acoustic noise. The amplitudes of micro30 sensors are about 20 dB higher than the amplitudes of WD sensors. This is because of higher sensitivity of micro30 sensor as compared to WD sensor.

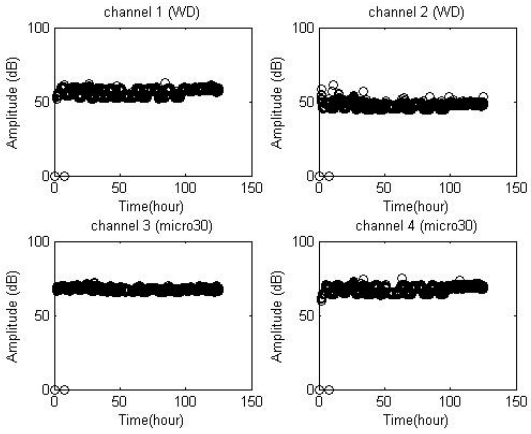


Figure 6. Amplitude histories of AE sensors over 130 hours testing.

The frequency spectrum is divided into three segments in order to find the energy distribution of each segment. The frequency ranges are 100-200 kHz (partial power 1), 200-300 kHz (partial power 2), and 300-400 kHz (partial power 3). It is predicted that the increase in partial power 3 with time (i.e., the frequency spectrum shifts towards to higher frequencies) may relate to active crack growth. This is based on the hypothesis that the crack emission has higher frequencies than acoustic noise due to operational conditions.

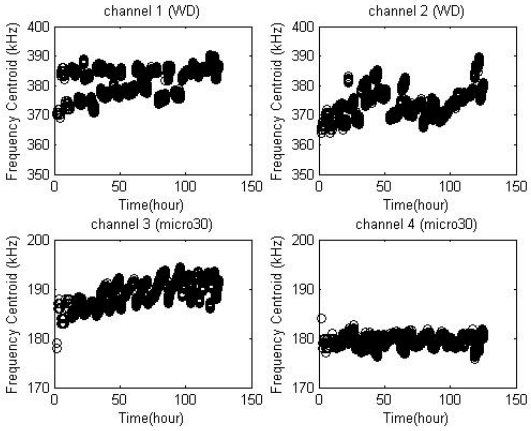


Figure 7. Frequency centroid histories of AE sensors over 130 hours testing.

The frequency centroid and partial power 3 histories of four sensors are shown in Figure 7 and Figure 8, respectively. The range of the frequency centroid values is controlled by the sensor type. For instance, the mean frequency centroid of WD sensors is about 380 kHz while it is near 180-190 kHz for micro30 sensors. The variations within the data set depend on the sensor position. While channels 3 and 4 are the same sensor type, there is no change in the features of channel 4 throughout testing. The WD sensors do not show any consistent variations as well. The interfaces and materials in the path of propagating elastic waves from the source to the sensor location influence the final surface motion that the sensor converts into electrical signal.

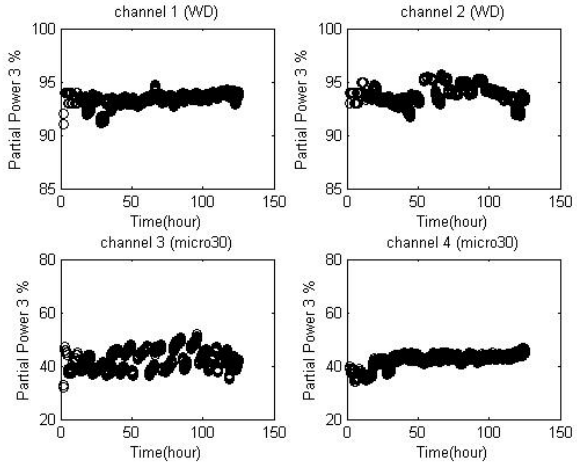


Figure 8. Partial power 3 histories of AE sensors

The comparison of different sensor types and positions indicates that the AE features depend on the selected sensor type and position relative the crack initiation.

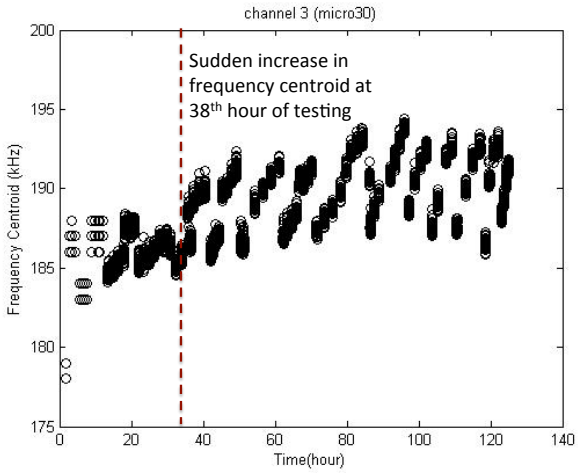


Figure 9. Frequency centroid history of channel 3 (micro30).

Figure 9 shows the frequency centroid history of channel 3. The AE data collection was continuous about 8 hours of each day. When the data was plotted, it is considered as continuous. The frequency centroid values were consistent until the 38th hour of testing. After this point of testing, it is observed that the frequency centroid is gradually increased after the initiation of each test. Based on the hypothesis of high frequency emissions due to active flaws, the 38th hour of testing may be considered as the initiation of active flaw or severe fretting damage on gearbox parts other than the splines. The predicted time of crack growth is in good agreement with the crack growth observed in the replica where crack size was measured at intermitted test intervals. It is important to note that the AE data at the beginning and end of each testing were not used in the analyses, as there were significant variations in the acoustic noise due to the gearbox operation.

3.3. Principal Component Analysis

The AE waveforms can be represented by various time dependent and frequency dependent features. Pattern recognition methods utilize the AE features as the descriptors of the multivariate analysis through mixing time domain and frequency domain features in order to differentiate source mechanisms. The pattern recognition methodology includes unsupervised and supervised modes. The unsupervised mode is applicable if there is no prior knowledge about classes (Anastassopoulos, and Philippidis 1994). The challenge of the unsupervised pattern recognition method is to define the physical meaning of each class that the method finds. In this study, five features, including absolute energy, frequency centroid, partial power 1 to 3 are selected, and principal component analysis is applied in order to perform multivariate analysis. Figure 10 shows the first to the fourth PCA histories of the channel 3 data. The third PCA has similar indication as the frequency centroid, while the fourth PCA has no sensitivity to the active flaw. Understanding the physical meaning of PCI components is an ongoing research problem.

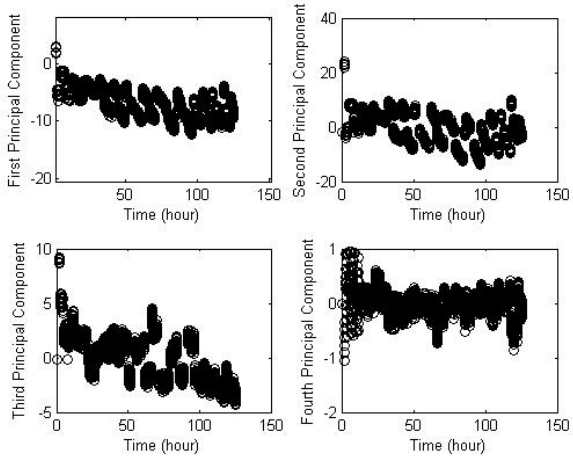


Figure 10. PCA values of channel 3 data using five features.

4. THE COMPARISON OF AE WITH GEARBOX OPERATIONAL VARIABLES

In addition to the AE data, several parameters related to the gearbox operation are collected simultaneously. Figure 11 and Figure 12 compare the frequency centroid history of channel 3 with the FWD load (one of the two moment drivers) and the temperature for the test period of 27 hours to 83 hours. The direction of the FWD and AFT loads were varied for each test to alternate the radial load vectors on the bearings. The restart points of two tests are highlighted in the figures. At the beginning of 27th hour of testing, the frequency centroid did not change with the load. A slight increase in the frequency centroid with the load direction is observed at the initiation of testing. However, the variation within the test data is consistent. The test initiation point can be selected as the reference point, or normalized data can be utilized for pattern recognition methods.

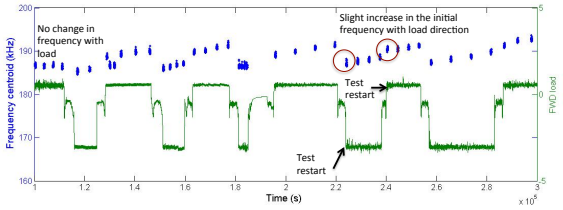


Figure 11. The comparison of AE data with gearbox temperature.

The gearbox temperature also does not influence the acoustic frequency. As shown in Figure 12, there is a slight increase in temperature at the initial part of the plot; however, the frequency centroid values stayed constant.

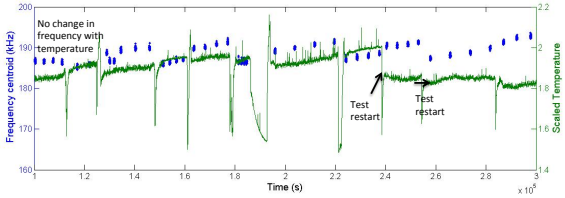


Figure 12. The comparison of AE data with gearbox FWD load.

If operational conditions influenced the AE features, the changes in the AE features due to crack and operational conditions should have been decoupled. This is very important to develop universal pattern recognition approach. Otherwise, operational variables such as temperature, forward load etc should be parts of variables influencing the patterns in the AE data.

5. DISCUSSION

The AE data recorded over 130 hours of gearbox operation show that time domain feature of amplitude does not change throughout testing when time-based data acquisition approach is implemented. Frequency domain features show variations in time while they are not influenced by the operational conditions of the gearbox. The estimated crack initiation time agrees well with the replica result where crack size was measured at different intervals of testing. The interfaces and materials in the path of propagating elastic waves from the source to the sensor location influence the final surface motion that the sensor converts into electrical signal. Therefore, pattern recognition method should be developed for specific sensor and position. If the geometry and materials of gearbox are modified, the AE features are influenced, and pattern representing crack growth becomes different.

6. CONCLUSION

In this study, the AE data was recorded during the initial crack growth from the notched spline, and recorded high frequency data in 5-second intervals for the entire 130 hours of gearbox testing. Four AE sensors (two different types) were mounted on the gearbox housing at different positions in order to understand the influences of sensor type/location and gearbox operational conditions to the AE characteristics. It is observed that the AE features extracted from the AE signals are influenced by the sensor type and location. As the pattern recognition methods rely on the AE features as the descriptors, they should be developed for a specific sensor type and position. The primary features sensitive to potential flaws are identified as the frequency domain features including frequency centroid and partial powers. The AE features are compared with the gearbox operational variables including FWD load and temperature. It is concluded that the operational variables have no significant influence on the frequency contents of the AE signals.

ACKNOWLEDGEMENT

This material is based upon work supported by the U.S. Naval Air Systems Command (NAVAIR) under Contract No. N68335-13-C-0417 entitled "Hybrid State-Detection System for Gearbox Components" awarded to the Metis Design Corporation. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NAVAIR.

REFERENCES

Anastassopoulos, A.A. and Philippidis, T.P. (1994). "Clustering Methodology for the Evaluation of Acoustic Emission from Composites," *Journal of Acoustic Emission*, Volume 13, pp. 11-22.

Chaari, F., Fakhfakh, T. and Haddar, M. (2009). "Analytical Modelling of Spur Gear Tooth Crack and Influence on Gearmesh Stiffness," European Journal of Mechanics A/Solids, Vol. 28, pp. 461-468.

Chen, Z. and Shao, Y. (2011). "Dynamic Simulation of Spur Gear with Tooth Root Crack Propagation along Tooth Width and Crack Depth," Engineering Failure Analysis, Vol. 18, pp. 2149-2164.

Dempsey, P.J. (2003). "Integrating Oil Debris and Vibration Measurements for Intelligent Machine Health Monitoring," NASA TM-2003-211307.

Eftekharnejad, B. and Mba, D.. (2011) "Monitoring Natural Pitting Progress on Helical Gear Mesh using Acoustic Emission and Vibration," *Strain*, Vol. 47, pp. 299-310.

Eftekharnejad, B., Addali, A. and Mba, D. (2012). "Shaft Crack Diagnostics in a Gearbox," *Applied Acoustics*, Vol. 73, pp. 723-733.

Gu, D.S., Kim, J.G., An, Y.S., and Choi, B.K. (2011). "Detection of Faults in Gearboxes using Acoustic Emission Signal," *Journal of Mechanical Science and Technology*, Vol. 25, No. 5, pp. 1279-1286.

Li, C.J., Lee, H. and Choi, S.H. (2002). "Estimating Size of Gear Tooth Root Crack using Embedded Modeling," Mechanical Systems and Signal Processing, Vol. 16, No. 5, pp. 841-852.

Li, R., and He, D. (2012). "Rotational Machine Health Monitoring and Fault Detection using EMD-Based Acoustic Emission Feature Quantification," *IEEE Transactions on Instrumentation and Measurement*, Vol. 61, No. 4, pp. 990-1001.

Li, R., Seckiner, S.U., He, D., Bechhoefer, E. and Menon, P. (2012). "Gear Fault Location Detection for Split Torque Gearbox using AE Sensors," *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, Vol. 42, No. 6, pp. 1308-1317.

Li, R., Seckiner, S.U., He, D., Bechhoefer, E. and Menon, P. (2012). "Gear Fault Location Detection for Split Torque Gearbox using AE Sensors," *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, Vol. 42, No. 6, pp. 1308-1317.

Loutas, T.H., Roulias, D., Pauly, E., and Kostopoulos, V. (2011). "The Combined use of Vibration, Acoustic Emission and Oil Debris on-line Monitoring towards a more Effective Condition Monitoring of Rotating Machinery," *Mechanical Systems and Signal Processing*, Vol. 25, pp. 1339-1352.

Ozevin, D., Godinez, V., Dong, J., and Carlos, M. (2007). "Damage Assessment of Gearbox Operating High Noisy Environment Using Waveform Streaming Approach," *Journal of Acoustic Emission*, pp. 355-363

Samuel, P.D. and Pines, D.J. (2005). "A Review of Vibration-based Techniques for Helicopter Transmission Diagnostics," Journal of Sound and Vibration, Vol. 282, pp. 475-508.

Wang, W. (2008). "An Enhanced Diagnostic System for Gear System Monitoring," IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 38, No. 1, pp. 102-112.

Yesilyurt, I., Gu, F. and Ball, A.D. (2003). "Gear Tooth Stiffness Reduction Measurement using Modal Analysis and its use in Wear Fault Severity Assessment of Spur Gears," *NDT&E International*, Vol. 36, pp. 357-372.

A Data Driven Method for Model Based Diagnostics and Prognostics

Michael D. Bryant [1]

[1,] *Mechanical Engineering, University of Texas at Austin, Austin, Texas 78712-0292*
bryantmd@austin.utexas.edu

ABSTRACT

This article's model based diagnostics system has four modules. Diagnosis and fault location forms physics models of the machine, measures states off the real in-service machine, generates simulated machine states and simulated sensor outputs for the machine model with loads same as the real machine, and compares simulated sensor outputs to real sensor outputs. The parameter tuning module adjusts (tunes) the parameters of the model until the simulated sensor outputs closely mimic real sensor outputs. Tuning transfers information on the system's health from the sensor data to the model's parameters. Parameters changed from nominal values locate faults and bad parts. For the health assessment module to assess machine health, we view a machine as a "machine channel" that organizes power and information flow through the machine. Machines focus power via an organization inherent in its components and design. Broken or degraded components disrupt this organization and the power and information flows. Shannon's information theory for communications channels can then be applied as a health metric to this "machine channel". Ageing of components degrades machine functional health. To prognose future health, differential equations that model ageing of the machine's components are formulated and solved. These equations predict component degradation, and update values of parameters in the model associated with component ageing. With these future parameter values, simulations of the machine operation model can then predict "future" machine behavior, and system health. This article demonstrates these methods on motors and a pump.

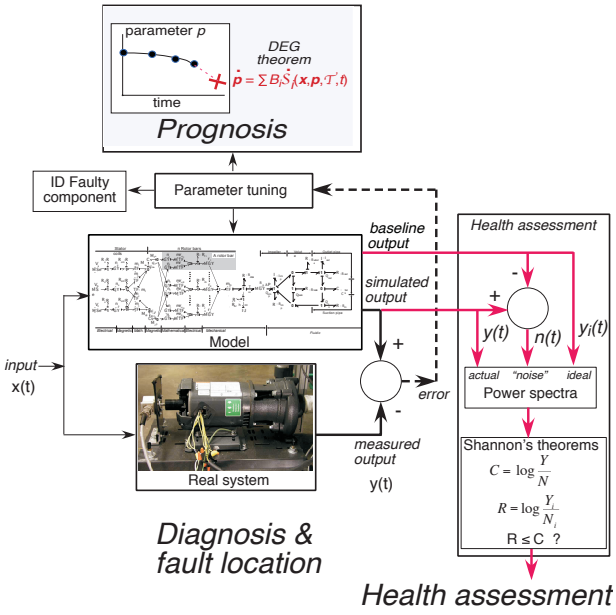


Figure 1. A schematic of the model based diagnostic system, consisting of four modules: diagnosis and fault location, consisting of real machine, inputs, sensor outputs, and physics model of machine; parameter tuning module to extract health condition from measurements; health assessment module to assess machine functional capability; and prognosis module to forecast future machine condition.

1. INTRODUCTION

A diagnostic system should detect, isolate, and identify the type and nature of a fault; determine the severity of the fault on system performance and the urgency of corrective action; analyze accommodation of the fault; and finally, forecast future behavior of the system, given the presence and future state of the fault. This article overviews a model based diagnostics and prognostics system, shown schematically in Fig. 1. The system integrates several modules developed at University of Texas at Austin into an overall diagnostics system. The modules described in the next section were all developed from fundamentals of physics and information theory.

Model-based diagnostics constructs models of machines to interpret sensor signals in terms of faults and locate and track faults in machines. Figure 1 depicts the system consisting of real machine; inputs to the machine; a physics based model of the machine with many physical states and parameters; outputs from the machine measured by sensors, and corresponding outputs simulated by the model; a module that tunes or adjusts the numerical values of the

Michael D. Bryant. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

model's parameters to make the model's simulated outputs mimic the real machine's measured outputs; a health assessment module to evaluate the system's health or ability to do a job using the measured signals; and a prognosis module which forecasts the changed values of parameters of an aged machine, via a thermodynamics based method of modeling effects of degradation. With these future "aged" parameters, the model can simulate future machine behavior to predict the future health condition of the machine. In the following sections, the components and operation of each module will be described in detail.

Since these modules are all based on fundamentals of physics and information theory, the reliability of this overall diagnostics system is extremely high.

2. MODEL BASED DIAGNOSTIC SYSTEM

Each module of Fig. 1 will be introduced and described.

2.1. Diagnostics and Fault Location Module (DFLM)

In Fig. 1, the Diagnostics & fault location module consists of a sensory system to observe the real machine and faults, and a detailed physics based model of the machine system to interpret the sensor signals. The model simulates the behavior of both machine and sensor system.

2.1.1. Sensor System and Observability

For any diagnostics system to work properly, the sensors must collect sufficient, correct and appropriate information from the system. The sensor system must be observable to the faults.

Model based diagnostics do not require exotic sensors. Simple and common sensors found on industry machines can usually ensure diagnosability. Although models interpret the sensor signals, these signals must contain sufficient information to enable a correct diagnosis. For motors, typically measured are voltages, currents, run-outs, speed, vibration and temperature by sensors such as potential/current transformer, hall-effect sensor, capacitive probe, encoder, accelerometer, and thermocouple. Key to selecting the right combination of sensors with enough information to detect a fault is fault observability, which in this context measures how well parameters can be inferred from information contained in error signals of model outputs and measurements (Analytic Sciences Corporation, 1974).

A dynamic system model is required to assess observability of a sensor system to any state or signal in a machine, such as a fault-induced signal. Nakhaeinejad & Bryant (2011) assessed observability to faults for an AC motor. Alternatively, sensitivity of sensor signals to changes in a fault can be studied, as Bryant, Nakhaeinejad & Choi (2011) did for the motor pump system presented in this article.

2.1.2. System Model

The model interprets the complex sensor signals. The model consists of differential equations that govern the physics of the machine. The model based diagnostic system of this article employs extremely detailed physics based models with direct physical correspondence between elements in the model and components and faults in the real machine. All relevant physics and effects are embedded in the model. Although this imbues the model with many degrees of freedom, many states, a high dynamic order, very many system parameters, and extreme nonlinearities, *this complexity is required in the model to interpret the equally complex sensor data*, which contains multiple competing signals from the many components and physical effects in a real machine. For example, in a motor, the bearing vibration signals measured by accelerometers are contaminated with vibrations from the motor's rotor reacting to harmonics of the magnetic field. These vibrations have harmonic components similar to the bearing, which confounds signal based bearing diagnostics.

During a simulation of the machine model, the model is given the same inputs as the real machine, see Fig. 1. Simulations attempt to emulate the real machine's dynamic states, up to and including the sensor measurements. Note the model contains a model of the sensor behavior. Signals measured off the real machine by sensors are then compared to corresponding signals derived from simulations of the model. For simulations to emulate real machine behavior, i.e., for the model's outputs to match the real machine's outputs, the model's parameters are tuned—adjusted until simulated outputs overlay measured outputs. This is the function of the parameter tuning module.

2.2. Parameter Tuning Module (PTM)

The parameter-tuning module accepts sensor signals from the real machine, and commands a simulation of the model. Initially, the model's parameter values are those of a healthy machine[1]. The simulation, given the same inputs as the real machine, computes system states up to and including the (simulated) sensor measurements. The parameter-tuning module subtracts the simulated sensor outputs from the corresponding measured sensor outputs, Fig. 1, and constructs an error function as the sum of the differences squared. Minimization of this error function drives an iterative process that corrects those parameters of the model associated with the known faults that compromise operation. Industry usually knows where and how faults occur in their machines, unknown is when the fault will occur. Parameter tuning performs simulations with updated parameters until the error function is within an acceptable tolerance. To

[1] These healthy machine parameters can be estimated via a combination of the machine's design specs and/or tuning of parameters using a baseline signal that exemplifies health.

reduce computational load, only tuned are those parameters associated with the machine's faults and ageing, which cause the measured signals to change.

If the model's parameters have a direct physical correspondence to components and faults in the real machine, tuning of parameters until simulations emulate real machine behavior extracts and puts the health condition information from the sensor signals into the parameter values of the model. Since the model's parameters have a direct physical correspondence to components and faults in the real machine, the tuned parameter values locate the fault and inform on its severity, via how much the parameter(s) have changed from nominal healthy values. If the model is physics based, the updated parameter values are easily interpreted in terms of physical effects of faults. This removes the pattern classification and training problem usually associated with heuristic and signal based diagnostic systems.

The parameter tuning module is challenged by the quality of the sensor data, which is compromised by noise and inadequate observability. Measurements inherently include sensor and physical process noise, and observability of a measurement can vary markedly if the system is nonlinear. To address these challenges, we tried online tuning with Kalman and Extended Kalman filters, and offline tuning with an algorithm that minimizes global errors A Kalman filter augments a physics model with a statistical model of the noise, for more accurate estimates of states (Haykin, 2001). Kalman filters first predict future states, and then correct these states recursively, using the error between simulation and measurement, and a Kalman gain, which arises from the analytical solution to the error minimization problem. For nonlinear systems, the extended Kalman filter includes the parameters to be tuned as extra components in the state vector. This usually results in a more nonlinear system, because the governing differential equations—the system differential equations augmented with equations that describe parameter degradation—usually involve products of parameters and states.

The Kalman filters operating with the detailed physics models described earlier operated satisfactorily in the presence of noise, but often failed due to observability issues associated with the nonlinear nature of the models. Sensors observability of faults can reduce and even vanish due to the nonlinearities of machine models (Nakhaeinejad & Bryant, 2011). A Kalman filter sequentially processes a signal point by point and must "latch on" to the signal. When extreme nonlinearities reduced sensor observability, the Kalman filter would detach from the signal, and become unstable. An offline tuning method was must less affected by this waning observability issue.

The offline tuning method (Rengarajan, 2010) constructs a multi-dimensional parameter space, with each parameter to be tuned assigned a coordinate axis. Thus N parameters require an N dimensional space, and tuning the set of parameters is tantamount to searching for the correct point in the space. The search is limited to those regions of the space where parameter values are physically possible or reasonable. First, a deterministic sampler scans the entire admissible region, without bias to any particular sub-region, using a grid. At each sampling point, error residuals between measured sensor signals and model simulated sensor signals are calculated to identify five regions where residuals are smallest. Then a "Non-Dominating Sorting Genetic Algorithm" is run in small regions about the five zones to pinpoint the global minimum. This algorithm involves randomness, to maximize the likelihood of attaining a global minimum in case the deterministic sampler gets stuck in local minima. The resulting global minimum values are ranked, and the top candidate is used as the system parameter values. Tuning is iterative and ends once error tolerances are met.

The offline tuner was tested on a DC motor where the created rotor bar resistance faults were known (Rengarajan, 2010). Tuned parameters included rotor inertia, motor constant, rotor bar resistance, and damping coefficient. Motor speed was varied by suitably adjusting the input voltage. The tuning algorithm estimated the rotor bar resistance values using motor speed measurements to within a few percent.

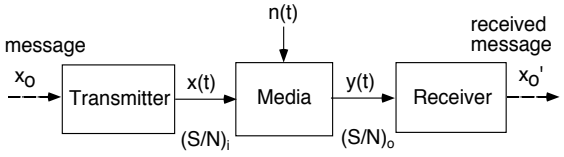


Figure 2. Shannon & Weaver (1948) communications channel.

2.3. Health Assessment Module (HAM)

The health assessment module determines the functional health capability of the machine, based on the channel capacity C from Shannon's information theory. Shannon's C is the maximum amount of information x_o in bits per second that can be transmitted through a channel contaminated with noise, but yet received without error. Shannon's theory, which specifies signal to noise power ratios Y/N and channel bandwidth ω, has underpinned all communication systems design since 1948. Obey Shannon's theorems and a system works, otherwise not.

The Shannon & Weaver (1948) channel capacity for a time continuous channel with white Gaussian noise in Fig. 2 is

$$C = \omega \log_2\left(\frac{Y}{N}\right) \tag{1}$$

which involves average power

$$Y = P\{y(t)\} = \frac{1}{T}\int_0^T \left[y(t)\right]^2 dt \tag{2a}$$

of signal $y(t) = x(t) + n(t)$, and power of noise $n(t)$,

$$N = P\{n(t)\} = \frac{1}{T}\int_0^T \left[n(t)\right]^2 dt \cdot \tag{2b}$$

In Fig. 2, the received signal $y(t)$ is the transmitted signal $x(t)$ corrupted with noise $n(t)$ from the channel. Here bandwidth ω (Hz) of the channel is usually determined via Nyquist's rules.

A machine will be viewed as a "machine communications channel" with input signals transmitted over a "machine channel" and received as the machine's output signals. Here faults create and add "fault noise" to output signals. To apply Shannon's fundamental theorems to assess machine health, noise will be defined as

$$n_i(t) = y(t) - y_i(t), \tag{3}$$

the difference between output $y(t)$ of the degraded machine, and a baseline signal $y_i(t)$ that exemplifies health, as discussed in Costuros & Bryant (2014). The noise signal of Eq. (3), a residual between degraded $y(t)$ and baseline $y_i(t)$, contains the "fault noise" signals generated by faults, and random sensor and system noise present in both $y(t)$ and $y_i(t)$. Of course, to use Eq. (3) in an industry setting, signals $y(t)$ and $y_i(t)$ must first be correlated in time to have the same starting point and be synchronized.

Applying Eqs. (2) to baseline signal $y_i(t)$ and noise $n_i(t)$ of Eq. (3) produces a channel capacity for the baseline signal

$$R = \omega_i \log_2\left(\frac{Y_i}{N_i}\right). \tag{4}$$

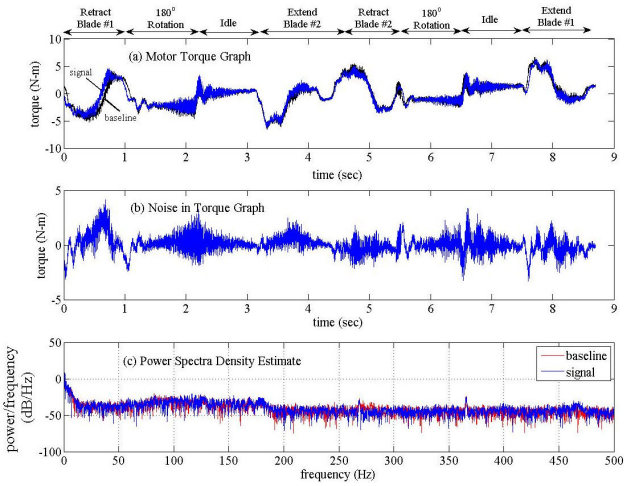


Figure 3. Motor torque response from robot 1 on 1/15/10.

Here bandwidth ω_i of baseline signal $y_i(t)$ is usually equal to ω. Equation (4) will be used in place of Shannon's rate of information in Shannon's test channel health, wherein if

$$R \le C, \tag{5}$$

the system will satisfactorily perform its function, otherwise not. Costuros (2013) showed that unless the power of sensor and system noise overwhelms (> 20%) the fault noise, the test of Eqs. (1)–(5) will work in an industry setting.

Costuros & Bryant (2014) demonstrated the efficacy of channel capacity as a health metric via tests on ageing industry robots, which will be reviewed here. The channel capacity technique was tested on eight DC motors in four industry robots, each initially in good operating condition. An identical sequence of voltage steps (transmitted channel inputs) were repetitively applied to all motors, and torque signals $y(t)$ (received channel outputs) were then collected from all motors. Motors ran continuously from 12/9/09 to 2/5/10. Motor output torques were measured on 12/9, 12/18, 1/15, 1/21 and 2/5. The 12/9 measurements were designated as baseline signals $y_i(t)$ exemplary of good health, to which all subsequent measurements $y(t)$ on the same motor were compared. Before any calculations, a signal $y(t)$ was first correlated to its $y_i(t)$ to synchronize signal alignments in time. Figure 3a shows robot 1 motor torque $y(t)$ on 1/15 (blue curve), and its baseline $y_i(t)$ (black curve). Fault noise in Fig. 3b obtained via Eq. (3) distills the fault induced signal from $y(t)$. Power spectra of signal $y(t)$ and noise $n_i(t)$ computed via Eq. (3) are in Fig. 3c. Channel capacity C was estimated via Eq. (1) and tabulated in Table 1.

Table 1: Channel capacity for motors of robots vs. time.

Date	Robot 1		Robot 2		Robot 3		Robot 4	
	Motor A	Motor B	Motor A	Motor B	Motor A	Motor B	Motor A	Motor B
12/18/09	2193	2164	1780	2326	1878	1647	2051	1679
1/15/10	1965	1784	1335	1481	1307	964	1383	989
1/21/10	2039	1827	1375	1466	1465	1072	1406	1005
2/5/10	1907	1985	1188	1340	1252	929	1475	1043
% change 12/18 - 1/15	10%	18%	25%	36%	30%	41%	33%	41%
	BEST 1	2	3	WORST 8	4	6	5	7
% change 12/18 - 1/21	7%	16%	23%	37%	22%	35%	31%	40%
	BEST 1	2	3	WORST 8	4	5	6	7
% change 12/18 - 2/5	13%	8%	33%	42%	33%	44%	28%	38%
	BEST 2	1	4	WORST 8	5	7	3	6

For measurements after 12/18, fractional changes in channel capacity $\%C = 1 - C/C_{12/18}$ relative to values for 12/18 measurements were tabulated in Table 1 for all motors. Inspection of the upper rows reveals a trend of diminishing channel capacity over time. For example, for motor B of robot 2, C diminishes from 2,326 to 1,340 from 12/18/09 to

2/5/10. In subsequent rows, the percent change of channel capacity from 12/18 to 1/21 is displayed, along with a composite of human produced evaluations of motion performance by a team of industry engineers and technicians. The human evaluations rank-ordered the motors and identified the best and worst performing motors. In general, the channel capacity estimates agreed well with human (team) assessments. Motor 'A' in robot 1, deemed BEST by the team, had the smallest channel capacity reductions. Motor 'B' in robot 2, rated WORST by the team, consistently showed the largest reduction of channel capacity and was prematurely removed from service due to development of a grinding noise. In general, the drop in the channel capacity values correlated very well with the human perceived amount of motor degradation. An overall decline in channel capacity indicates degradation. This application suggests that the channel capacity metric can quantify system degradation in industry settings. The channel capacity decreases in Table 1 are not strictly monotonic. Fluctuations in the C values in Table 1 for most motors at the beginning of tests are consistent with a break-in process, wherein performance does vary. For these motors, the majority of faults occurred on the motor bearings due to lubrication breakdown.

2.4. Prognosis Module (PM)

The prognosis module, schematically shown at the top of Fig. 1, forecasts future values of the model's parameters via differential equations that govern the ageing and degradation of the system's components. These equations and the ageing phenomena typically have time constants much larger than the characteristic times of the machine in operation. To make the Prognosis module compatible with the other diagnostic modules, the component degradation equations are posed in terms of those system parameters P_k that change due to component degradation. This degradation or ageing worsens the faults. Equations that govern degradation (Bryant, 2014) can be formulated via the Degradation Entropy Generation theorem (Bryant, Khonsari & Ling, 2008), which equates the rate of change of a variable w that measures the degradation (i.e., monotonically increases or decreases as the fault becomes more severe) to a linear combination of the irreversible entropies S_i' generated by the n dissipative processes underlying the degradation, i.e.,

$$\frac{dw}{dt} = \sum_{i=1}^{n} B_i \frac{dS'_i}{dt} . \qquad (6a)$$

Equation (6a) is founded on the laws of thermodynamics. Although the B_i constants are usually unknown, the irreversible entropies S_i' on the right side of Eq. (6a) can be formulated in terms of the power dissipated by components, divided by a temperature associated with the degradation, using knowledge of the mechanics of dissipation losses and the ageing and degradation mechanisms. If degradation changes parameter P_k then $P_k = P_k(w)$, and via the chain rule $dP_k/dt = dP_k/dw\ (dw/dt)$. Substitution of Eq. (6a) gives

$$\frac{dP_k}{dt} = \sum_{i=1}^{n} \left(B_i \frac{dP_k}{dw} \right) \frac{dS'_i}{dt} = \sum_{i=1}^{n} B_i^* \frac{dS'_i}{dt} \qquad (6b)$$

where dP_k/dw was grouped with the constants B_i to form new constants B_i^*. Values for these constants can be obtained via the tuning module, since a history of values for parameters P_k will be available from past tunings of the operational model to sensor data.

Over the course of multiple tunings, a record of the parameter's values P_k versus time can be constructed, as in the graph seen in the Prognosis section of Fig. 1. Future values of parameters P_k, associated with faults could be forecast by fitting a curve through the record of P_k data points, and extrapolating that curve into the future, as in point "X". A more accurate forecast uses Eqs. (6b) and tunes the unknown constants B_i^* with the record of P_k versus time. Then using the most recent value of P_k as an initial condition, the P_k can be forecast much further into the future. With future values for the parameters P_k, the machine model shown in Fig. 1, given the machine's inputs, can now simulate the future degraded machine behavior and its output signals $y(t)$. With these future output signals $y(t)$ inserted into Eq. (3), the health assessment module can assess future machine performance.

2.5. Diagnostic System Operation

The diagnostic system operates as follows. Abbreviations are defined in the headings of section 2.

1) DFLM simulates the model of Fig. 1 with inputs same as the service loads on the real machine, and outputs including the sensor states.
2) DFLM compares simulated "sensor" signals to the real sensor measured signals.
3) PTM adjusts (tunes) the model's parameters, until simulated sensor readings overlay real sensor readings. Accuracy is a few percent. The tuned model now emulates machine behavior, and distilled into the tuned parameter values is the machine's health condition.
4) PTM detects and locates faults by tracking changes in the numerical values of the tuned parameters. Larger changes implies a more severe fault(s).
5) HAM compares the machine's signals $y(t)$ to a baseline signal $y_i(t)$ that exemplifies machine health, and assesses machine condition by calculating the machine channel capacity C, and the percent change from baseline channel capacity.
6) PM with the history of the model's parameters from past tunings, solves the differential equations governing parameter change, and predicts future parameter values.

7) DFLM simulates the model of the "future" machine with inputs same as past service loads on the real machine, and outputs "simulated sensor" states to predict future machine operation.
8) HAM compares the "future" machine signals $y(t)$ to the baseline signal $y_i(t)$, and calculates the channel capacity of the future machine to assess future machine condition.

3. MOTOR PUMP APPLICATION

The techniques discussed in section 2 will be demonstrated on a centrifugal pump driven by an induction motor, Fig. 4. Faults introduced include extra resistance in the motor's stator circuit and blockage in the pipe following the pump.

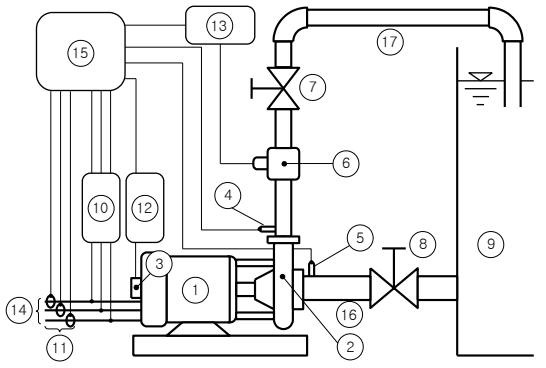


Figure 4. Motor-pump system test setup.

3.1. Motor Pump Model

Within the **DFLM** module in Fig. 1, in the block labeled "model" is a bond graph model of the dynamics of a squirrel cage induction motor driving a centrifugal pump. From the bond graph, differential equations governing motor-pump operation were extracted and presented in Bryant & Choi (2012). The model has parameters with nominal values listed in Table 2.

In Fig. 4, a 3-phase, 2 hp, 3600 rpm squirrel cage induction motor (1) drives a centrifugal pump (2) (19 m max. head). Measured are 3 phases of input voltage (10), 3 phases of currents (11) via Hall effect sensors, motor rotational speed (3), flow rate at the outlet pipe (6), and pressures at inlet (5) and outlet (4) of the pump via pressure transducers.

Table 2 Parameters of motor-pump, with nominal (healthy system) values.

Parameters	Description	Healthy value
R_s	Stator coil resistances (Ω)	1.0281
R_{sm}	Stator magnetic losses (1/Ω)	366.7
$R_{r1},\ldots,R_{r34}$	Rotor bar resistance (Ω)	0.8663
L_s	Stator inductances (H)	0.1033
L_r	Rotor inductances (H)	0.1377
L_m	Mutual inductances (H)	0.1162
R_{br}	Mechanical friction (N-s/m)	0.0034
R_{disk}	Mechanical friction (N-s/m)	1.1e-5
R_{imp}	Loss in impeller (kg/m^7)	3.6e11
R_{volute}	Loss in volute (kg/m^7)	7.0e9
R_{leak}	Leakage loss (kg/m^7)	1.6e15
R_{out}	Loss in outlet pipe (kg/m^7)	2.3e11
R_{in}	Loss in inlet pipe (kg/m^7)	1.0e10
J	Moment of inertia (N-m^2)	0.003802
I_{imp}	Liquid inertia in impeller (kg/m^5)	8.6e7
I_{out}	Liquid inertia in outlet pipe (kg/m^5)	2.5e6
n_s	Number stator coil turns	111
n_r	Number rotor coil turns	1
r_{i1}	Impeller inner radius (m)	0.025
r_{i2}	Impeller outer radius (m)	0.05
B_{i1}	Axial width at impeller inlet (m)	0.01
B_{i2}	Axial width at impeller outlet (m)	0.01
β_1	Blade angle at impeller inlet (°)	15
β_2	Blade angle at impeller outlet (°)	30

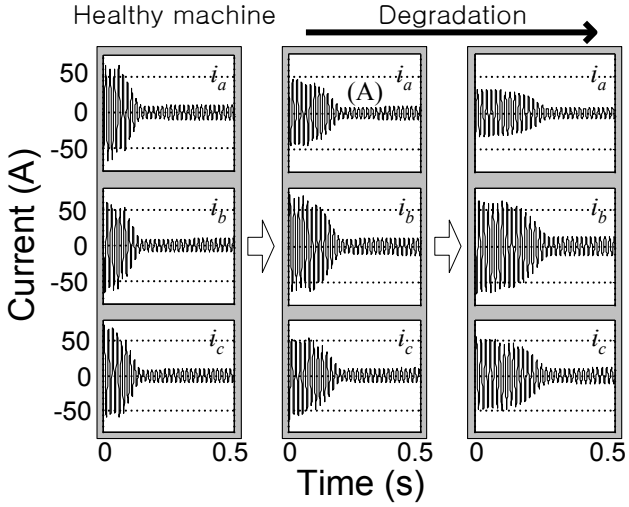


Figure 5. Currents in (a) healthy motor, and with extra resistance (b) 2.5 Ω and (c) 4.5 Ω in phase *a* of stator.

For the stator circuit fault, Fig. 5 shows the change of measured 3 phase currents (*a*, *b*, *c*), from healthy to degraded. The (b) and (c) subfigures in Fig. 5 connected 2.5 Ω and 4.5 Ω in series to the *a* phase stator coil. As the resistance fault increases, the time to steady state increases, and magnitudes of i_a reduce. Higher resistance

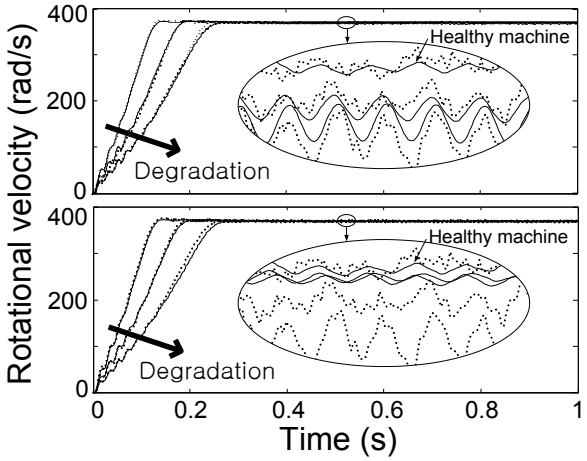


Figure 6 Measured (dotted lines) and tuned (solid lines) rotational velocity by stator coil resistances (upper) and by motor inductances (bottom).

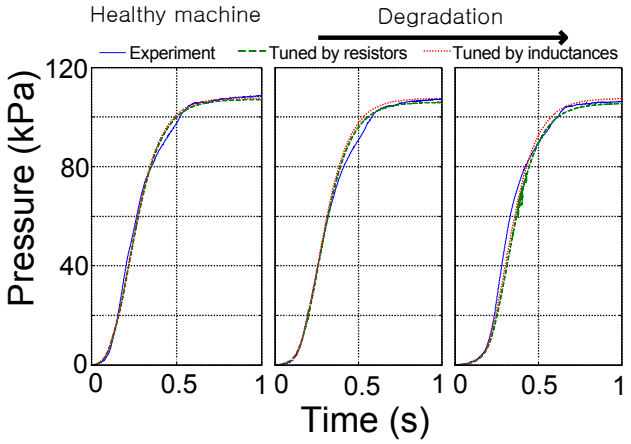


Figure 7 Pressures for a) healthy, (b) 2.5 Ω, (c) 4.5 Ω.

simultaneously affected measured current, rotational velocity, and pressure, Figs. 6 and 7.

Table 3 assesses sensitivity of measured states to changes in selected parameters, as substitute for an observability assessment of the sensor system. After each parameter in table 2 was individually perturbed 1% of nominal value, a simulation was performed to observe changes in system response. The number of '+' symbols in any row in table 3 indicates the influence of each parameter's change. Measured currents, rotational velocities, and pressures are sensitive to changes in stator coil resistances (R_{sa}, R_{sb}, R_{sc}) or motor inductances (L_s, L_r, L_m), even though the origin of the fault is the stator resistance R_{sa}. First, the motor-pump model was tuned by adjusting stator coil resistances only, and tuned a second time by adjusting motor inductances only. The error function for tuning was the sum of the square of differences between measured and simulated rotational velocity. Currents and pressures were not considered in the error function. Simulations of healthy (Table 2) and degraded machines (Table 4 presented in Figs. 6, 7, and 8) nearly overlay experiments. Although Figures 7 and 8 tuned parameters so that rotational velocity simulations overlaid measurements, as a by-product, current and pressure simulations also overlaid their respective measurements.

Table 3 Sensitivity of system states to 1% change in parameters.

Parameters	Sensitivities		
	Rotational speed	Currents	Pressure (Flow rate)
R_s	++	+++	++
$R_{r1},\ldots,R_{r34}$	.	+	.
L_s, L_r, L_m	+++	+++	+++
R_{br}, R_{disk}	++	+	+
R_{imp}	.	.	++
R_{out}	.	.	++
R_{in}, R_{volute}, R_{leak}	.	.	.

Simulations with parameters tuned by stator coil resistances and by motor inductances gave similar rotational velocities (Fig. 6) and pressures (Fig. 7). However, the magnified details shown in the bubbles in Fig. 6 of rotational velocities at steady state suggests that simulations from tuning by stator coil resistances more closely fits measurements, than tuning by motor inductances, for the resistance fault. Since the induction motor model represents a symmetrical electric machine, each of R_{sa}, R_{sb}, and R_{sc} with the tuned values can in turn produce the rotational velocities in Fig. 6. The magnitude of currents i_a in Fig. 5 reduce most as the value of connected resistor R_{sa} increases. Other currents (i_b and i_c in Fig. 5) change only little. Thus R_{sa} has to be the largest among the tuned resistances. Fig. 8 compares simulated to measured current i_a (Fig. 5), after assigning the largest value

Table 4 Parameter tuning data.

Parameters		Healthy value	Connected resistor	
			2.5 (Ω)	4.5 (Ω)
Tuning by resistances	R_{sa} (Ω)	1.0281	2.0525	5.0668
	R_{sb} (Ω)		1.0959	1.3719
	R_{sc} (Ω)		0.5296	1.3931
Tuning by inductances	$L_{s\alpha}$ (H)	0.1033	0.1037	0.1041
	$L_{s\beta}$ (H)		0.1031	0.1037
	$L_{r\alpha}$ (H)	0.1377	0.1382	0.1387
	$L_{r\beta}$ (H)		0.1379	0.1382
	$L_{m\alpha}$ (H)	0.1162	0.1152	0.1143
	$L_{m\beta}$ (H)		0.1154	0.1148

Subscripts a, b, c, α, and β denote magnetic axes.

of tuned stator coil resistance to R_{sa}.

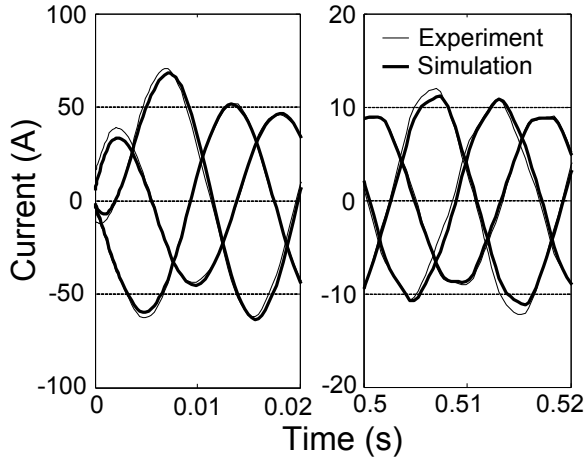


Figure 8. Magnified view of current (A) in Fig. 5 with tuned response after adjusting stator coil resistances.

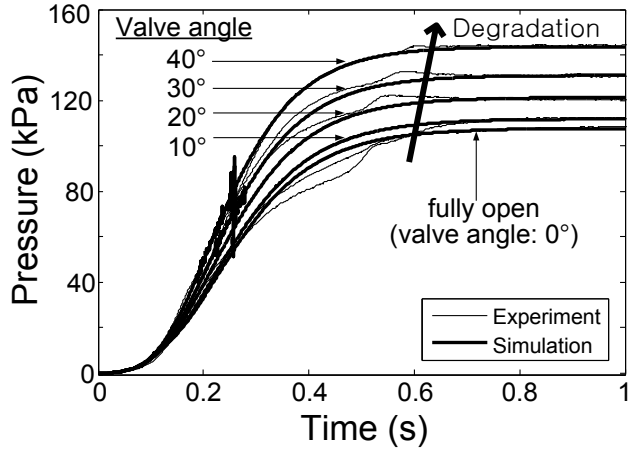


Figure 9. Tuned pressures by hydraulic loss at outlet pipe, R_{out}.

Fluid loss, R_{out} in the centrifugal pump model models pipe line losses such as friction loss, expansion loss, contraction loss, valve loss, etc. The butterfly valve (7) of Fig. 4 in the middle of the outlet pipe was closed in 10° increments to mimic increasing resistance. The valve can be adjusted from fully open 0° to fully closed 90°. Closing the valve from 0° to 40° had little effect on measured currents and rotational velocity, but pressure signals increased significantly. From Table 3, R_{out} was selected as the parameter for tuning, since it increases outlet pressure significantly, with little effect on currents and rotational velocity. R_{imp} was deselected, since increasing R_{imp} decreases outlet pressure. Figure 9 shows the measured pressure as valve angle changed from 0° to 40°, and the simulated pressure obtained by adjusting R_{out} from $2.3x10^{11}$, to $2.4\ x10^{11}1$, $2.7\ x10^{11}$, $3.1\ x10^{11}$, and $3.3\ x10^{11}$ (kg/m^7). Changing R_{out} had negligible effect on current and rotational velocity, as implied by Table 3.

The channel capacity C for measured outputs of stator phase current i_a and motor speed ω were calculated via Eq. (1) and presented versus resistance in stator phase a in Fig. 10. Values were normalized by maximum values, so the largest C value is one. As the fault worsens and system performance degrades as shown in Figs. 5 and 6, the channel capacity monotonically diminishes, similar to that of Table 1.

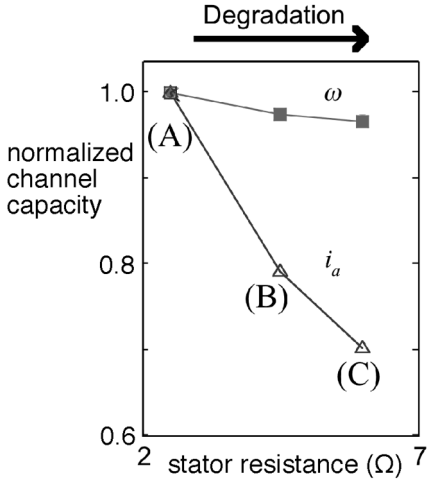


Figure 10. Channel capacity vs. stator a resistance.

4. CONCLUSION

A model-based diagnostic system was presented, with application to a motor-pump. Physics models of high detail and fidelity permitted simulations to match experiments with marginal error. Parameter tuning selected values of parameters such that simulations overlaid measurements. Contained in the tuned values of parameters is the machine health condition. The channel capacity health metric assessed fault severity. For signals over channels through a machine that possess observability of the fault(s) in question, this article shows that models and parameter tuning can locate and isolate faults. For signals observable to a given fault, channel capacity monotonically diminished with severity of the fault.

ACKNOWLEDGEMENT

Research reported in this publication was supported in part by the National Science Foundation (NSF) grant IIP 1266279. The content is solely the responsibility of the

authors and does not necessarily represent the official views of the National Science Foundation.

REFERENCES

Analytic Sciences Corporation. Technical Staff & Gelb, A., (1974). *Applied Optimal Estimation*. Cambridge, MA: M.I.T. Press.

Bryant, M.D., (2014). Modeling Degradation Using Thermodynamic Entropy, submitted to PHM 2014 Conference.

Bryant, M.D., Khonsari, M.M. & Ling, F.F. (2008). On the thermodynamics of degradation, *Proceedings of Royal Society of London Series A*, vol. 464 (2096), pp. 2001-2014, doi:10.1098/rspa.2007.0371.

Bryant, M.D., & Choi, J.H. (2012). Model based fault diagnostics of induction motor and centrifugal pump, *Proceedings of MFPT 2012 Conference*. April 24- 26, Dayton, OH.

Bryant M.D., Nakhaeinejad M. & Choi J., (2011). Model based diagnostics and fault assessment of induction motors with incipient faults, *Proceedings of the Society for Experimental Mechanics Series*, vol. 8, pp. 439 – 449.

Costuros T., (2013). *Application of Communication Theory to Health Assessment, Degradation Quantification, and Robot Cause Diagnosis*, Doctoral dissertation, University of Texas at Austin, Austin, TX, https://repositories.lib.utexas.edu/bitstream/handle/2152/21566/COSTUROS-DISSERTATION-2013.pdf.

Costuros, T. & Bryant, M.D. (2014). Application of information theory's channel capacity as an industry machine health and diagnostic metric. *Proceedings of MFPT 2014 Conference*. May 19-22, Virginia Beach, VA.

Haykin, S. S. (2001). *Kalman Filtering and Neural Networks*. New York: Wiley.

Nakhaeinejad, M. & Bryant, M. D., (2011). Observability Analysis for Model-Based Fault Detection and Sensor Selection in Induction Motors, Journal of Measurement Science and Technology, vol. 22(7), pp. 075202.

Rengarajan, S.B., (2010). A Method for Parameter Estimation and System Identification for Model Based Diagnostics Masters Thesis, University of Texas at Austin, Austin, TX.

Shannon, C.E. & Weaver, W., (1948). *The Mathematical Theory of Communication*, Urbana, IL: The University of Illinois Press.

Shannon C. E. (1949). Communication in the presence of noise, *Proceedings of the IRE*, vol. 37, pp. 10 – 21.

BIOGRAPHY

Michael D. Bryant, Accenture Endowed Professor of Mechanical Engineering at University of Texas at Austin, Austin TX, USA, was born in Danville IL on Feb. 8, 1951. His education includes B.S., Bioengineering, University of Illinois at Chicago, Chicago, IL, USA, 1972; graduate study in Information Engineering, University of Illinois at Chicago, 1972-1974; M.S., Mechanical Engineering, Northwestern University, Evanston, IL, USA, 1980; and Ph.D., Engineering Science and Applied Mathematics, Northwestern University, 1981. He has been with University of Texas at Austin since 1988. From 1981-1988 he was an Assistant and Associate Professor at North Carolina State University. His interests include tribology, mechatronics, manufacturing, and system modeling. He is a fellow of American Society of Mechanical Engineers, was Editor in Chief of ASME Journal of Tribology from 2005-2012, and is a member of Institute of Electrical and Electronics Engineers and Sigma Xi.

Planetary Gearbox Fault Diagnosis Using a Single Piezoelectric Strain Sensor

Jae Yoon[1], David He[2], Brandon Van Hecke[3], Thomas J. Nostrand[4], Junda Zhu[5], and Eric Bechhoefer[6]

[1,2,3]*University of Illinois at Chicago, Chicago, IL, 60607, USA*
jyoon52@uic.edu
davidhe@uic.edu
bvanhe2@uic.edu

[4,5]*Renewable NRG Systems, Hinesburg, VT, 05461, USA*
tjn@renewablenrgsystems.com
jz@renewablenrgsystems.com

[6]*Green Power Monitoring Systems, Essex Junction, VT, 05452, USA*
eric@gpms-vt.com

ABSTRACT

Planetary gearboxes are widely used in the drivetrain of helicopters and wind turbines. Any planetary gearbox failure could lead to breakdown of the whole drivetrain and major loss of helicopters and wind turbines. Therefore, planetary gearbox fault diagnosis is an important topic in prognostics and health management (PHM). Planetary gearbox fault diagnosis has been done mostly through vibration analysis over the past years. Vibration signals theoretically have the amplitude modulation effect caused by time variant vibration transfer paths due to the rotation of planet carrier and sun gear, and therefore their spectral structure is complex. It is difficult to diagnose planetary gearbox faults via vibration analysis. Strain sensor signals on the other hand have less amplitude modulation effect. Thus, it is potentially easy and effective to diagnose planetary gearbox faults via stain sensor signal analysis. In this paper, a research investigation on planetary gearbox fault diagnosis via strain sensor signal analysis is reported. The investigation involves using time synchronous average technique to process signals acquired from a single piezoelectric strain sensor mounted on the housing of a planetary gearbox and extracting condition indicators for fault diagnosis. The reported investigation includes analysis results on a set of seeded fault tests performed on a planetary gearbox test rig in a laboratory. The results have showed a satisfactory planetary gearbox fault diagnostic performance using strain sensor signal analysis.

Jae Yoon *et al.* This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Gearboxes are widely used in almost every powertrain of rotating systems such as automobile, helicopter, wind turbine, and etc. According to Link *et al.* (2011), approximately 59% of the failure modes in wind turbines involved gear failures. Astridge *et al.* (1989) indicated that 19.1% of all the helicopter transmission failures came from the gear failure. Gearbox failures are normally accompanied by unexpected increment in operation cost and catastrophic disaster followed by loss of life. Especially, the planetary gearbox (PGB) is one of the most critical components in generating uplift force in a helicopter transmission system and converting wind power to electrical power in a wind turbine drive train system. However, the fault detection of planetary gearbox is very complicate since the complex nature of dynamic rolling structure of planetary gearbox does not allow for direct attachment of sensors within the rotating elements. A large portion of planetary gearbox diagnostic system has been devoted to vibration analysis using accelerometers. A vibration analysis technique namely "vibration separation" was introduced by McFadden & Howard (1990), Howard (1990), and McFadden (1991). Vibration separation enables to decompose a raw vibration signal into multiple PGB component (*e.g.* sun, planet, or ring) oriented vibration signals by taking windowed vibration signals only when the vibration sensor, ring gear, planet gear, and sun gear are aligned inline. The windowed vibration signals are recombined specifically for the targeted gear component by utilizing the geometric properties of corresponding PGB. Subsequent studies by McFadden (1994), Samuel *et al.* (2004), and Lewicki *et al.* (2011) validated this research with slightly modified versions of the

technique. However, the fundamental idea of vibration separation remains unchanged. Wu *et al*. (2004) have shown the detectability of planet carrier crack in a planetary gearbox. In their study, raw vibration data and time synchronous average (TSA) data were transferred to frequency domain and wavelet domain to obtain differentiable features. In a paper by Patrick *et al*. (2007), a vibration data based framework for on-board fault diagnosis and failure prognosis of helicopter transmission component was presented. In their study, TSA pre-processed vibration data and particle filter based diagnostic and prognostic models were presented. Yu *et al*. (2010) compared a raw vibration signal and TSA signal with a wavelet transformed vibration signal to obtain desirable fault feature. Bartelmus & Zimroz (2009) showed that the spectral characteristics of vibration signal obtained from planetary gear help not only fault detection but also gear fault location. Feng & Zuo (2012) derived mathematical models of faulty planetary gear for detecting and locating fault by considering characteristic frequency of amplitude modulation (AM) and frequency modulation (FM) effects.

In a recent paper, Feng & Zuo (2013) pointed out that vibration signals theoretically have the amplitude modulation effect caused by time variant vibration transfer paths due to the unique dynamic structure of rotating planet gears. Therefore, it is difficult to diagnose PGB faults via vibration analysis. One attractive solution to this problem is to use alternative sensor signals that have less sensitivity to AM effect for PGB fault diagnosis and prognosis. Feng & Zuo (2013) have shown the effectiveness of torsional vibration analysis for PGB fault diagnosis using a torque sensor. The frequency characteristics of torsional vibration were shown to be solely sensitive to the AM and FM effects caused by gear faults under constant torque on input and output shafts. Kiddy *et al.* (2011) used fiber optic strain signals for PGB fault diagnosis and showed a close relationship between strain measurement and torque changes. Even though promising, the research reported in the literature on using less AM effect sensitive signals for PGB fault diagnosis has certain limitations. The torque sensors used by Feng and Zuo (2013) are more expensive than vibration and strain sensors and require special installation. The fiber optic strain sensor array used by Kiddy *et al*. (2011) had to be embedded in the PGB in order to be effective. The strain signals of fiber optic strain sensor can only be sampled at a maximum sampling rate up to 1 kHz, which limits its coverage on shaft speed above 2060 rpm. Also in Kiddy *et al*. (2011), the strain signals were analyzed the same way as vibration signals. Fiber optic sensor signals were analyzed using vibration separation technique after low frequency components were filtered out. No effective signal analysis techniques have been developed for strain signals. Piezoelectric (PE) strain sensor is desirable in having an improved strain resolution and applicability of higher sampling rate in comparison with the conventional strain gauge sensors (Banaszak 2001) or the fiber optic strain sensors (Jiang *et al.* 2014).

To overcome the above mentioned challenges in developing effective PGB fault diagnosis capability, a research investigation on planetary gearbox fault diagnosis via strain sensor signal analysis has been conducted and is reported in this paper. The PE strain sensors based planetary gearbox fault diagnosis method can be considered as an attractive alternative to traditional vibration analysis based approaches. A key characteristic of PE materials is the utilization of the direct piezoelectric effect to sense structural deformation and the converse piezoelectric effect to actuate structures. Compared to the conventional strain gauge sensors and accelerometers, the PE strain sensors have certain advantages that could be summarized as follows: (1) ability to measure the first derivative of physical deformation with less sensitive AM and FM effect , (2) high linearity and sensitivity from their superior noise immunity as compared to differentiated sensing performance of conventional strain sensors (Lee & O'Sullivan, 1991, Banaszak 2001), (3) high frequency range (Jiang *et al.* 2014), (4) space-efficiency without a structural change on the measuring target (Kon *et al*. 2007), and (5) negligible high temperature effect on the measurement output (Sirohi & Chopra, 2000, Jiang *et al.* 2014). The aforementioned benefits allow for PE strain sensors to potentially have greater sensing resolution and accuracy.

The remainder of the paper is organized as follows. Section 2 gives a detailed explanation of the proposed methodology. In Section 3, the details of the seeded fault tests on a laboratory planetary gearbox test rig and the experimental setup used to validate the proposed methodology are provided. Section 4 presents the planetary gearbox fault diagnosis results from the seeded fault tests. Finally, Section 5 concludes the paper.

2. METHODOLOGY

An overview of the proposed methodology is provided in Figure 1. First, the PE strain sensor signals and tachometer signals are digitized simultaneously. Then, a band pass filter is applied so that the band passed signals could contain the information related to the planetary gearbox conditions. Using the tachometer signals, the TSA signals can be obtained along with residual signal and energy operator (EO). Residual signal is the TSA signal with shaft and mesh frequencies being removed and EO is a type of residual of the autocorrelation function (Teager, 1992).

In a related research on rotating machinery diagnostics, it has been shown that a deliberately chosen band pass filter improves diagnostic performance by removing shaft imbalance (Shiroishi *et al*., 1997). Thus, a band pass filter with low frequency bandwidth (*i.e.,* low pass filter) was applied to get the information associated with the gearbox condition while high frequency noises could be removed.

The major components of the methodology are explained in the following two sections. Section 2.1 provides a brief review of TSA and the computation of condition indicators (CIs) used for planetary gearbox fault diagnosis is explained in Section 2.2.

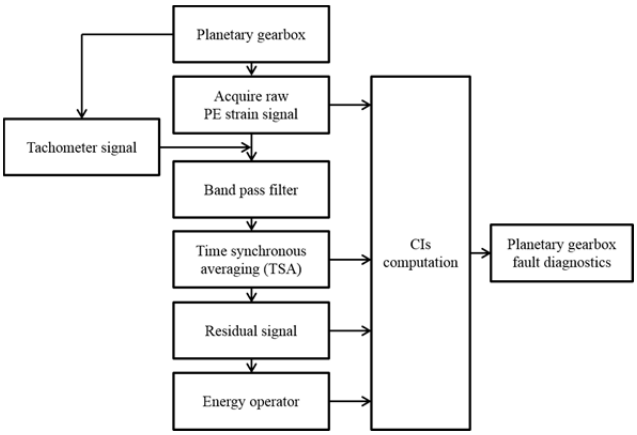


Figure 1. Overview of the methodology.

2.1. Time Synchronous Average

TSA is one of the most widely utilized signal processing techniques to extract a periodic waveform from noisy signals of rotating machines. The underlying idea of TSA is to obtain a periodically repeated waveform of interest over N number of revolutions. Theoretically, when a rotating machine is running at a constant speed, the periodic waveform is intensified while any noises are suppressed with a noise reduction rate of $\frac{1}{\sqrt{N}}$.

Consider a signal $x(t)$ composed of a periodic signal $y(t)$ with known period T_R and additive noise $e(t)$:

$$x(t) = y(t) + e(t) \tag{1}$$

Assuming the total number of N observed periods, the TSA of $x(t)$ can be expressed as:

$$a(t) = \frac{1}{N}\sum_{r=0}^{N-1} x(t - rT_R) \tag{2}$$

As $N \to \infty$, the TSA signal $a(t)$ approaches to $y(t)$. More details about TSA could be found in (Braun, 1975; McFadden, 1987; Bechhoefer and Kingsley, 2009).

Basically, TSA chops up the raw sensor signal into multiple single revolution signals. Then, each revolution signals are resampled (via stretching or shrinking) so as to have same sample points in one revolution. Then, the final periodic signal is obtained by averaging the resampled signals. After TSA is computed, any kind of fault detection condition indicators can be evaluated. Two major types of TSA techniques have been reported in the literature: TSA with tachometer as a reference signal and tachometer-less TSA. Since comparing those two techniques is beyond the scope of this paper, only the TSA with tachometer will be addressed herein. Even though successful TSA applications to many types of signals such as vibration and acoustic emission (AE) signals have been reported in the literature (Mcfadden, 1987; Bonnardot *et al*., 2005; and Qu *et al*., 2014), application of TSA to PE strain signal processing for planetary gear fault diagnosis has not yet been reported.

2.2. CIs for Planetary Gearbox Fault Diagnosis

Table 1 provides the definitions of the CIs investigated for PGB fault diagnosis. The CIs can be defined into five general types: root mean square (*RMS*), peak to peak (*P2P*), skewness (*SK*), kurtosis (*KT*), and crest factor (*CF*). Each type of CI can be computed using different input signals. In addition to TSA signals, other types of input signals can be generated: residual, narrow band (NB), AM, and FM. Residual is a TSA signal with the primary meshing and shaft components removed. The energy operator (EO) introduced by Teager (1992) is defined as the residual of the autocorrelation function as following:

$$x_{EO,i} = x_i^2 - x_{i-1} \cdot x_{i+1}, \quad (\text{for } i = 2, 3, \ldots, N-1) \tag{3}$$

where $x_{EO,i}$ is the i^{th} element of EO data; x_i is the i^{th} element of the input data x_{IN}. NB signals could be obtained by applying a narrow band pass filter on the TSA data. The width of the narrow band can be selected based on the gear fault frequency. In this paper, three narrow bands are selected based on sun gear fault frequency, planet gear fault frequency, and ring gear fault frequency, respectively. Finally, AM and FM signals are obtained by amplitude modulation and phase modulation of the narrow band filtered data.

3. EXPERIMENTAL SETUP

This section covers the experimental setup used to validate the PE strain sensor based planetary gearbox fault diagnostic technique. Figure 2 displays the planetary gearbox test rig used to collect the PE strain sensor data under different gear health and operating conditions.

3.1. The Planetary Gearbox Test Rig

The planetary gearbox test rig composes four main parts: (1) the data acquisition (DAQ) system, (2) the driving motor, (3) the gearbox, (4) the load generator. The DAQ system includes a National Instruments' DAQ board with a maximum analog input sampling rate of 1.25 MHz, a PE strain sensor, and a signal conditioner from PCB Piezotronics. The driving motor is a 3-phase 10HP induction motor with a motor controller. A Hall effect sensor was used as the tachometer paired with a toothed

Table 1. The definitions of the CIs.

		Input Signal (x_{IN})					
		TSA	Residual	EO	NB	AM	FM
CI	Description / Equation	Time synchronous averaged signal (x_{TSA})	TSA signal with the primary meshing and shaft components removed (x_{Res})	Energy operator: a residual of the autocorrelation function (x_{EO})	Narrow band pass filtered (x_{NB})	Amplitude modulation of NB filtered signal ($AM(x_{NB})$)	Frequency modulation of NB filtered signal ($FM(x_{NB})$)
Root mean square (*RMS*)	$RMS(x_{IN}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	$RMS(x_{IN})$: measures the magnitude of a discretized signal.					
Peak to peak (*P2P*)	$P2P(x_{IN}) = \frac{(\max_{1\le i\le N}(x_i) - \min_{1\le i\le N}(x_i))}{2}$	$P2P(x_{IN})$: measures the maximum difference within the data range.					
Skewness (*SK*)	$SK(x_{IN}) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^3}{\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\right]^3}$	$SK(x_{IN})$: measures the asymmetry of the data about its mean value. A negative *SK* value and positive *SK* value imply the data has a longer or fatter left tail and the data has a longer or fatter right tail, respectively.					
Kurtosis (*KT*)	$KT(x_{IN}) = \frac{N\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left[\sum_{i=1}^{N}(x_i - \bar{x})^2\right]^2}$	$KT(x_{IN})$: measures the peakedness, smoothness, and the heaviness of tail in a data set.					
Crest factor (*CF*)	$CF(x_{IN}) = \frac{P2P(x_{IN})}{RMS(x_{IN})}$	$CF(x_{IN})$: measures the ratio between $P2P(x_{IN})$ and $RMS(x_{IN})$ to describe how extreme the peaks are in a waveform.					

Note: x_i is i^{th} element of the input data x_{IN}; N is the length of the input data x_{IN}; max(·) returns the maximal element of input data x_{IN}; min(·) returns the minimal element of input data x_{IN}; $\bar{x}$ is a mean value of the input data x_{IN} defined as $\sum_{i=1}^{N} x_i / N$; NB, AM, and FM refers to a narrow band, amplitude modulation, and frequency modulation, respectively.

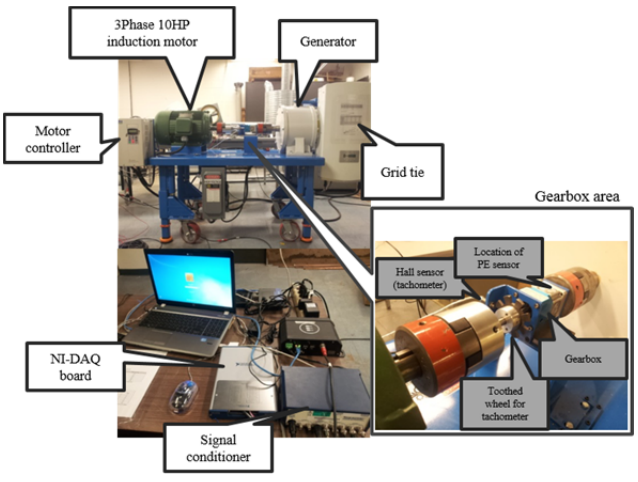


Figure 2. The planetary gearbox test rig for wind turbine simulator.

wheel mounted on the motor shaft. The output shaft of the gearbox is connected to a generator and a grid tie to serve as a load generator. The structure of the PGB test rig is similar to those used in a wind turbine. In th is study, a commercially available single stage planetary gearbox with a 5:1 speed reduction ratio was used. In Figure 3, a notional sketch of the planetary gearbox structure is p rovided. Amongst the three different planetary gearbox types, a specific planetary gearbox with standstill ring gear was used in this paper. For this type PGB, the number of teeth is linear to the radius of each gears pitch circle. This indicates that the gear ratio is also related to the angular velocity (ω) of the gears. The gear ratio can be defined as:

$$R = \frac{\omega_1}{\omega_A} = 1 + \frac{z_3}{z_1} \quad (4)$$

where ω_i is the angular velocity of the i^{th} gear component; z_i is the number of teeth on the i^{th} gear component; the gear component index subscripts 1, 2, 3, and *A* correspond to sun gear, planet gear, ring gear, and arm (*i.e.* planet carrier), respectively. The planet carrier rotation speed (*i.e.* output shaft speed) in frequency could be obtained as:

$$f_a = \frac{f_1}{R} \tag{5}$$

where f_i is the rotation speed in frequency at the i^{th} gear component. Also, a meshing characteristic frequency of planetary gearbox can be obtained as:

$$f_{12} = f_{23} = \frac{f_1 z_1 z_3}{(z_1 + z_3)} = \frac{f_1 \cdot z_3}{R} \tag{6}$$

where f_{ij} is the relative rotation speed in frequency between the i^{th} and j^{th} gear component.

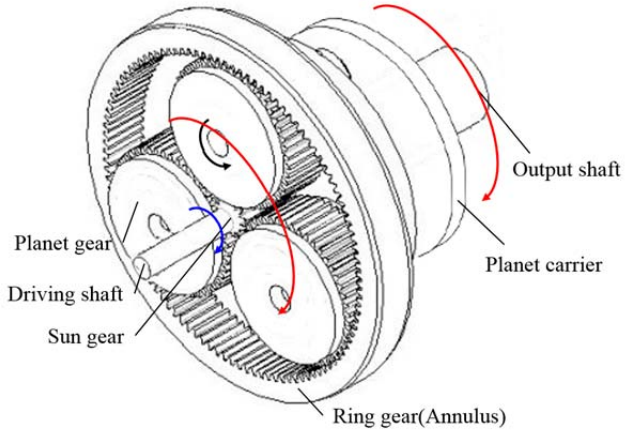


Figure 3. Notional sketch of the planet gearbox structure.

The most common three failure modes of a planetary gearbox are: sun gear fault, planet gear fault, and ring gear fault. Their corresponding fault frequencies are represented as follows:

$$f_{f,1} = s \cdot (f_1 - f_a) = \frac{f_1 z_3 s}{(z_1 + z_3)} \tag{7}$$

$$f_{f,2} = 2(f_2 + f_a) = \frac{4 n_1 z_1 z_3}{(z_3^2 - z_1^2)} \tag{8}$$

$$f_{f,3} = s \cdot f_a = \frac{f_1 z_1 s}{(z_1 + z_3)} \tag{9}$$

where $f_{f,i}$ represents the fault frequency at the i^{th} gear component; s represents the number of planet gears in the gearbox. For more details, see (Bartelmus and Zimroz, 2011). Tables 2 and 3 present the structural information and characteristic frequencies of the planetary gearbox used in this study.

Table 2. The parameters of the planetary gearbox

Parameter	Number of teeth on sun gear (z_1)	Number of teeth on planet gear (z_2)	Number of teeth on ring gear (z_3)	Number of planet gears (s)
Value	27	41	108	3

Table 3. Characteristic frequencies of the planetary gearbox at varied input shaft speed.

Input shaft speed (f_1)	Output shaft speed (f_a)	Meshing frequency ($f_{12} = f_{23}$)	Sun gear fault frequency ($f_{f,1}$)	Planet gear fault frequency ($f_{f,2}$)	Ring gear fault frequency ($f_{f,3}$)
10	2	216	24	10.67	6
20	4	432	48	21.33	12
30	6	648	72	32	18
40	8	864	96	42.67	24
50	10	1080	120	53.33	30

* All the values are in unit of Hz.

3.2. Seed Gear Faults

Three types of planetary gearbox faults were created: sun gear tooth fault, planet gear tooth fault, and ring gear tooth fault. Each type of the gear fault was created by artificially damaging a tooth on a sun gear, planetary gear, and rig gear, respectively (see Figure 4).

During the seeded fault tests, PE strain signals were collected with a sampling rate of 100 kHz. The tachometer signals were simultaneously recorded along with the PE strain signals to get revolution stamps. Both the healthy gearbox and the gearboxes with seeded faults were tested at 5 different input shaft speeds: 10 Hz, 20 Hz, 30 Hz, 40 Hz, and 50 Hz. At each speed, five samples were collected. In addition to the shaft speed variation, varying loading conditions were applied at the output shaft of the gearbox: 0%, 25%, 50%, and 75% of the maximum torque of the planetary gearbox. At each loading condition, 25 samples (five samples per shaft speed for 5 speeds) were taken. In addition, the PE strain sensors were mounted at the same location of the gearbox for each data collection.

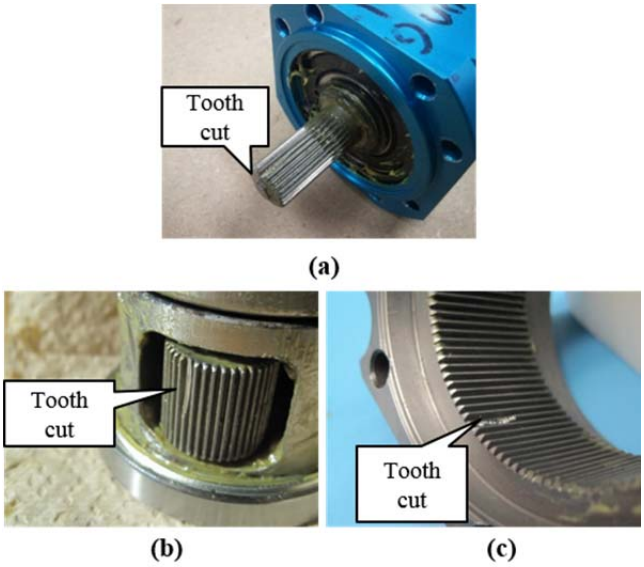


Figure 4. Seeded faults: (a) sun gear fault, (b) planet gear fault, (c) ring gear fault.

4. RESULTS

The validation results for the seeded fault tests conducted on the planetary gearbox test rig are provided in this section. Figure 5 shows a sample of the PE strain sensor signal and tachometer signal at 10Hz shaft speed for a duration of 0.3 seconds. Since the toothed wheel associated with the tachometer in the test rig has eight teeth, each input shaft revolution results in 8 zero crossings.

Before the TSA was computed, a band pass filter with a bandwidth of 1 Hz to 18 kHz was applied to the signals.

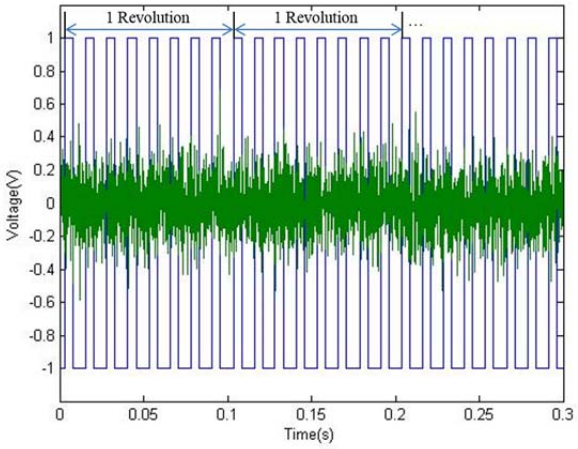


Figure 5. Sample of the healthy PE strain sensor signal and tachometer signal at 10Hz shaft speed.

Samples of the TSA signals of the PE strain sensor are provided in Figures 6 through 8. Figure 6 shows the TSA samples of the healthy gearbox with 50% loading at different shaft speeds. Figure 7 shows TSA samples with a shaft speed of 30Hz at different loading conditions. In Figure 8, TSA samples for different gearbox health conditions with shaft speed fixed at 30 Hz and loading at 50% are provided.

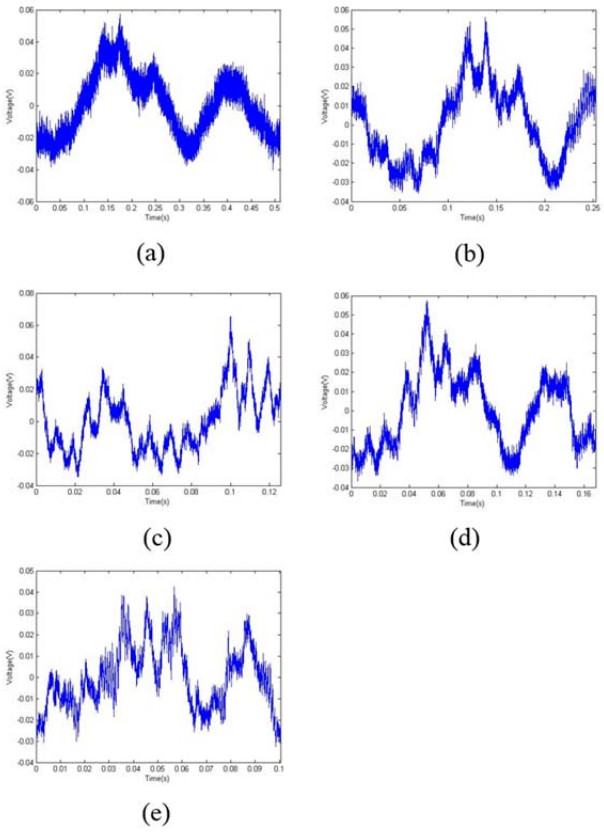


Figure 6. Samples of PE strain sensor signals of the healthy gearbox at different shaft speeds: (a) 10 Hz, (b) 20 Hz, (c) 30 Hz, (d) 40 Hz, (e) 50 Hz.

Once the TSA signals were obtained, then all of the CIs described in Section 2.4 were computed. Among the computed CIs, four of them were found effective: TSA RMS, TSA P2P, residual RMS, and residual P2P.

Figure 9 shows the TSA RMS plots for different gearbox health conditions at different shaft speeds and loading conditions. As one can see from Figure 9, by using TSA RMS alone, the three gear faults can be clearly separated. As the loading increases, the separation of the gear faults gets better. Also, by using TSA RMS alone, all the three gear faults can be clearly separated from the healthy condition. The detectability of the gear faults gets better as the loading increases. For all the 4 gearbox conditions, noted from Figure 9, the TSA RMS remains relatively stationary within the same loading condition regardless the change of the shaft speed. This shows that the PGB gear fault diagnostic capability of the TSA RMS is heavily

affected by the torque level of the gearbox. The vertical bar for each data poi nt shown in Figure 9 represents a 95% confidence interval of the estimated TSA RMS m ean. In order to ch eck the statistical significance of the gear fault separation using TSA RMS, analysis of variance (ANOVA) test was conducted using the TSA RMS data. In this test, it was assumed that the sha ft speed has no effect on TSA RMS within a loading condition.

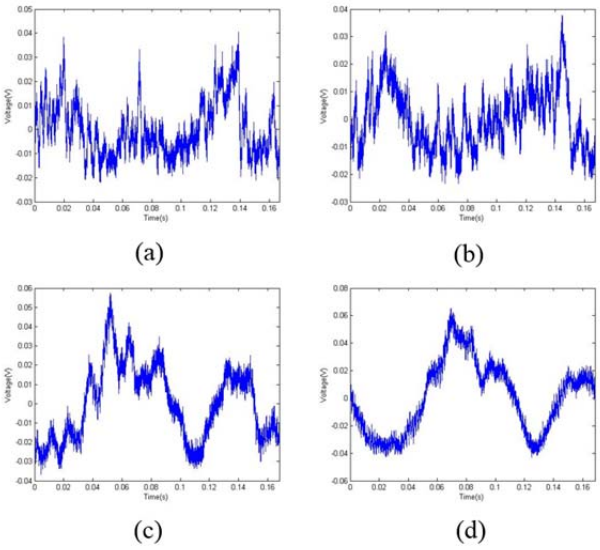


Figure 7. Samples of the PE strain sensor signals at different loading conditions: (a) 0%, (b) 25%, (c) 50%, (d) 75%.

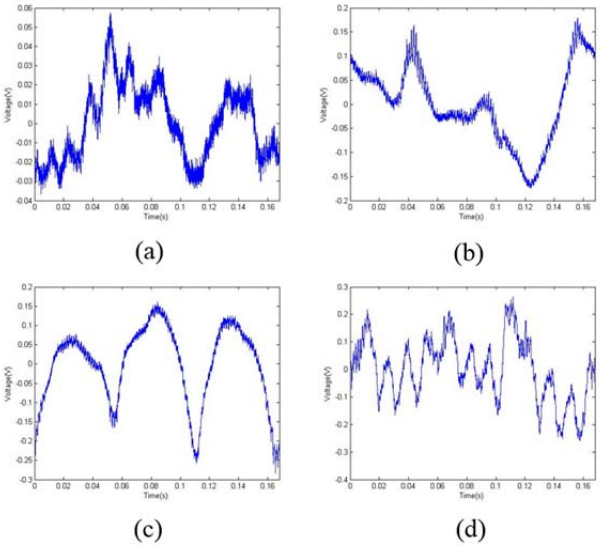


Figure 8. Samples of the PE strain sensor signals of different gearbox conditions: (a) healthy gearbox, (b) sun gear fault, (c) planet gear fault, (d) ring gear fault.

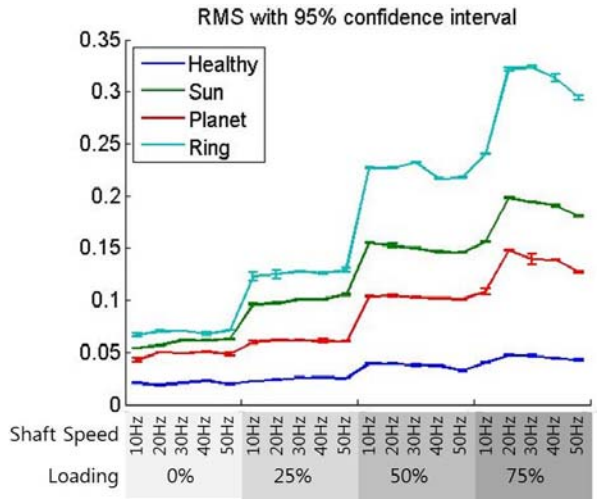


Figure 9. TSA RMS plots .

The following hypotheses were e stablished based on aforementioned assumptions:

$$\begin{aligned} &\mathrm{H}_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 \\ &\mathrm{H}_1: \text{at least one } \mu_i \neq \mu_j \\ &(\text{for } i, j = 1,2,3, \text{and } 4;\ i \neq j) \end{aligned} \tag{10}$$

where μ_i is mean TSA RMS of the i^{th} gear health condition at a fix ed loading condition, i = 1, 2, 3, and 4 rep resents healthy gearbox, sun gear fault, planet gear fault, and ring gear fault, respectively. Table 4 shows the summ ary of ANOVA results with a 99% confidence level.

From Table 4, *P*-values for all loading conditions are 0.000. With a 99% confidence level, the null hypotheses should be rejected ($\alpha = 0.01 > 0$). Therefore, it is safe to say that the separation of all the gear faults tested using TSA RMS is statistically significant at all loading conditions.

Table 4. Summary of ANOVA results for TSA RMS.

Loading	Source	DF	SS	MS	*F*	*P*
0%	Factor	3	0.0334141	0.0111380	1605.12	0.000
	Error	96	0.0006662	0.0000069		
	Total	99	0.0340802			
25%	Factor	3	0.1481272	0.0493757	8261.04	0.000
	Error	96	0.0005738	0.0000060		
	Total	99	0.1487010			
50%	Factor	3	0.4641124	0.1547041	10614.42	0.000
	Error	96	0.0013992	0.0000146		
	Total	99	0.4655116			
75%	Factor	3	0.845794	0.281931	781.55	0.000
	Error	96	0.034630	0.000361		
	Total	99	0.880424			

The results for other three CIs: TSA P2P, residual RMS, and residual P2P are presented in the same way as TSA RMS in the following. The resulting plots of the CIs are provided in Figures 10 to 12 and the ANOVA results in Tables 5 to 7, respectively.

Similar results like TSA RMS can be observed for other two CIs: TSA P2P and residual RMS. However, the diagnostic performance of these two CIs at 0% loading condition is not as good as TSA RMS. A clear diagnosis of the gear faults can be observed at 25%, 50%, and 75% loading conditions. When the loading level reaches 25% or above, TSA P2P and residual RMS can be ranked like TSA RMS as the following order: ring gear fault -> planet gear fault -> sun gear fault -> healthy gear. For residual P2P, a clear diagnosis of the gear faults can be observed only when the loading level reaches to 50% or above.

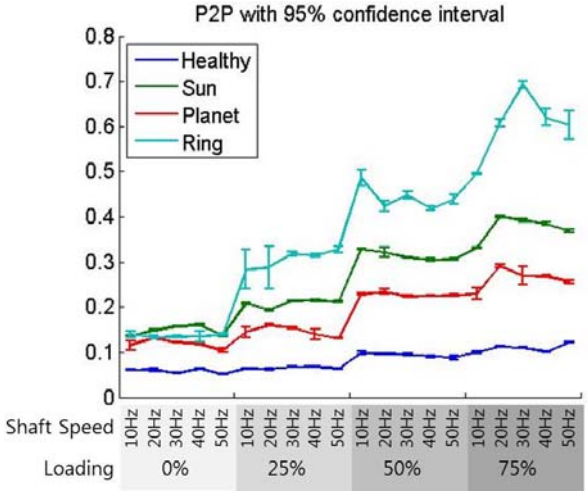


Figure 10. TSA P2P plots.

Table 5. Summary of ANOVA results for TSA P2P.

Loading	Source	DF	SS	MS	*F*	*P*
0%	Factor	3	0.1199638	0.0399879	611.06	0.000
	Error	96	0.0062822	0.0000654		
	Total	99	0.1262461			
25%	Factor	3	0.775791	0.258597	1065.47	0.000
	Error	96	0.023300	0.000243		
	Total	99	0.799091			
50%	Factor	3	1.615071	0.538357	2682.91	0.000
	Error	96	0.019264	0.000201		
	Total	99	1.634335			
75%	Factor	3	3.25105	1.08368	787.88	0.000
	Error	96	0.13204	0.00138		
	Total	99	3.38309			

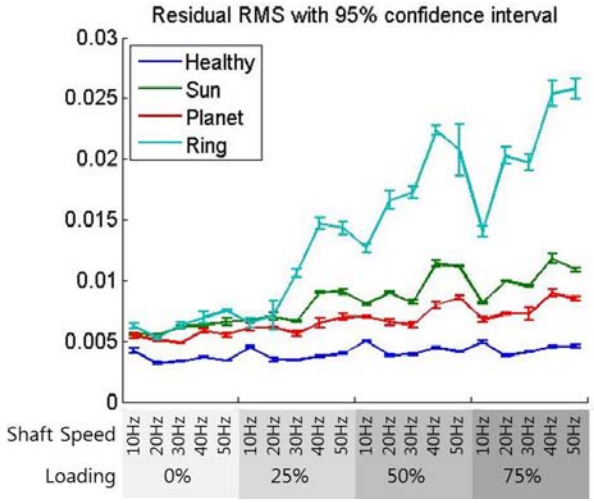


Figure 11. Residual RMS plots.

Table 6. Summary of ANOVA results for residual RMS.

Loading	Source	DF	SS	MS	*F*	*P*
0%	Factor	3	0.0001227	0.0000409	147.50	0.000
	Error	96	0.0000266	0.0000003		
	Total	99	0.0001493			
25%	Factor	3	0.0006061	0.0002020	56.46	0.000
	Error	96	0.0003436	0.0000036		
	Total	99	0.0009497			
50%	Factor	3	0.0025676	0.0008559	219.08	0.000
	Error	96	0.0003750	0.0000039		
	Total	99	0.0029427			
75%	Factor	3	0.0038871	0.0012957	233.04	0.000
	Error	96	0.0005337	0.0000056		
	Total	99	0.0044208			

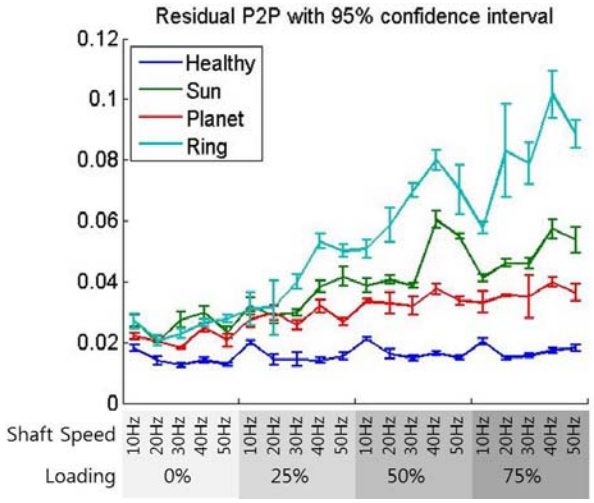


Figure 12. Residual P2P plots.

Table 7. Summary of ANOVA results for residual P2P.

Loading	Source	DF	SS	MS	*F*	*P*
0%	Factor	3	0.0019954	0.0006651	76.63	0.000
	Error	96	0.0008333	0.0000087		
	Total	99	0.0028287			
25%	Factor	3	0.0087545	0.0029182	79.85	0.000
	Error	96	0.0035084	0.0000365		
	Total	99	0.0122630			
50%	Factor	3	0.0323371	0.0107790	193.51	0.000
	Error	96	0.0053475	0.0000557		
	Total	99	0.0376846			
75%	Factor	3	0.0557005	0.0185668	239.39	0.000
	Error	96	0.0074456	0.0000776		
	Total	99	0.0631462			

Note that in Tables 5 to 7, even under the low loading conditions, the null hypothesis in (10) is rejected. This is because all the faulty CIs are significantly different from the healthy CIs even though the difference among the faulty CIs is not statistically significant.

5. CONCLUSIONS

In this paper, a new piezoelectric strain sensor based planetary gearbox fault diagnostic methodology was presented. The presented method was accomplished through a combination of band pass filtering, time synchronous average, and condition indicators to extract diagnostic features for planetary gear box diagnosis. First, the PE strain sensor signal is band pass filtered so as to retain the information related to the gear conditions. Then, TSA signal is computed to obtain the periodically repeated waveform while white noise is suppressed. The presented method was validated using data collected from seeded fault tests conducted on a planetary gearbox test rig in a laboratory. The validation results have shown that, by utilizing the TSA based PE strain sensor signal processing approach, fully separable diagnostic CIs towards all planetary gearbox fault types were captured regardless of shaft speed and output shaft loading condition. The current planetary gearbox diagnostic methods mainly rely on vibration signal analysis. They provide limited fault diagnosis for planetary gearboxes. The PE strain sensor based diagnostic technique presented provides an attractive alternative to the current vibration analysis based approach.

REFERENCES

Astridge, D. G. (1989). Helicopter transmissions - design for safety and reliability, *Proceedings of Mechanical Engineers: Journal of Aerospace Engineering*, Vol. 203, No. 2, pp. 123 - 138.

Bartelmus, W., & Zimroz, R. (2009). Vibration condition monitoring of planetary gearbox under varying external load, *Mechanical Systems and Signal Processing*, Vol. 23, No. 1, pp. 246 - 257.

Bartelmus, W., & Zimroz, R. (2011). Vibration spectra characteristic frequencies for condition monitoring of mining machinery compound and complex gearboxes, *Scientific Papers of the Institute of Mining, University of Technology, Studies and Research*, Vol. 133, No 40, pp. 17 - 34.

Banaszak, D. (2001). Comparison of piezoelectric strain sensors with strain gages, *Proceedings of the Annual Meeting of the American Statistical Association*, Atlanta, GA, Aug 5 – 9.

Bechhoefer, E. & Kingsley, M. (2009). A review of time synchronous average algorithm, *Annual Conference of the Prognostics and Health Management Society, Sep 27 – Oct 1*, San Diego, CA.

Bonnardot, F., Badaoui, M. E., Randall, R. B, Daniere, J., & Guillet, F. (2005). Use of the acceleration signal of a gearbox in order to perform angular resampling (with limited speed fluctuation), *Mechanical Systems and Signal Processing*, Vol. 19, No. pp. 766 - 785.

Braun, S. (1975). The extraction of periodic waveforms by time domain averaging, *Acustica*, Vol. 32, No. 1, pp. 69 - 77.

Feng, Z., & Zuo, M. J. (2012). Vibration signal models for fault diagnosis of planetary gearbox, *Journal of Sound and Vibration*, Vol. 331, No. 22, pp. 4919 – 4939.

Feng, Z. & Zuo, M. J. (2013). Fault diagnosis of planetary gearboxes via torsional vibration signal analysis, *Mechanical Systems and Signal Processing*, Vol. 36, No.2, pp. 401 - 421.

Howard, I. M. (1990). Epicyclic transmission fault detection by vibration analysis, *Australian Vibration and Noise Conference: Vibration and Noise - Measurement, Prediction, and Control*, Melbourne, Australia, pp. 171 - 178.

Jiang, X., Kim, K., Zhang, S., Johnson, J., & Salazar, G. (2014). High-temperature piezoelectic sensing, *Sensors*, Vol. 14, No. 1, pp.144 – 169.

Kiddy, J. S., Samuel, P. D., Lewicki, D. G., LaBerge, K. E., Ehinger, R. T., & Fetty, J. (2011). Fiber optic strain sensor for planetary gear diagnostics, *NASA Technical Report: NASA/TM-2011-217123*, NASA Glenn Research Center, Cleveland, OH.

Kon, S., Oldham, K., & Horowitz, R. (2007). Piezoresistive and piezoelectric MEMS strain sensors for vibration detection, *Proceedings of the SPIE: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 6529, pp. 1 - 11.

Lee, C. K. & O'Sullivan, T. (1991), Piezoelectric strain rate gages, *Journal of the Acoustical Society of America*, Vol. 90, No.2, pp. 945 - 953.

Lewicki, D. G., Laberge, K. E., Ehinger, R. T., & Fetty, J. (2011). Planetary gearbox fault detection using vibration separation technique, *NASA Technical Report:*

NASA/CR-2004-213068, NASA Glenn Research Center, Cleveland, OH.

Link, H., LaCava, W., van Dam, J., McNiff, B., Sheng, S., Wallen, R., McDade, M., & Lambert, S. (2011). Gearbox reliability collaborative project report: findings from phase 1 a nd phase 2 t esting," *NREL Technical Report: NREL/TP-5000-51885*, National Renewable Energy Laboratory, Golden, CO, USA.

McFadden, P. D. (1987). A revised model for the extraction of periodic waveforms by time domain averaging, *Mechanical Systems and Signal Processing*, Vol. 1, No. 1, pp. 83 - 95.

McFadden, P. D., & Howard, I. M. (1990). The detection of seeded faults in an epicyclic gearbox by signal averaging of th e vibration, *Aeronautical Research Laboratory*, Melbourne, Victoria, Australia.

McFadden, P. D . (1991). A technique for calculating the time domain averages of the vibration of the individual planet gears and the sun gear in an epicyclic gearbox, *Journal of Sound and Vibration*, Vol. 144, No. 1, pp. 163 - 172.

McFadden, P. D. (1994). Window functions for the calculation of the time domain averages of the vibration of the individual planet gears and s un gear in an epicyclic gearbox, *Journal of Vibration and Acoustics*, Vol. 116, No. 2, pp. 179 - 187.

Patrick, R., Orchard, M. E., Zhang, B., & Koelemay, M. D. (2007). An integrated approach to helicopter planetary gear fault diagnosis and failure prognosis, *Proceedings of IEEE AUTOTESTCON*, Baltimore, MD, pp. 547 - 552.

Qu, Y., He, D., Yoon, J., VanHecke, B., & Bechhoefer, E. (2014). Gearbox tooth cut fault diagnostics using acoustic emission and vibration sensors - a comparative study, *Sensors*, Vol. 14, No. 1, pp. 1372 - 1393.

Samuel, P. D ., Conroy, J. K ., & Pines, D. J. (2 004). Planetary transmission diagnostics, *NASA Technical Report: NASA/CR-2004-213068*, NASA Glenn Research Center, Cleveland, OH.

Shiroishi, J., Li, Y. Liang, S., Kurfess, T., & Danyluk, S. (1997). Bearing condition diagnosis via vibration and acoustic emission m easurement, *Mechanical Systems and Signal Processing*, Vol. 11, No. 5, pp. 693 - 705.

Sirohi, J. & Chopra, I. (2000). Fundamental understanding of piezoelectric strain se nsors, *Journal of Intelligent Material Systems and Structures*, Vol. 11, No. 4, pp. 246 - 257.

Teager, H. M ., & Tea ger S. M. (1992). Evidence for nonlinear sound production mechanisms in the vocal tract, in *Speech Production and Speech Modeling Symposium, Time Frequency and Time-Scale Analysis*, edited by Hardcastle, W. J. & March al, A., Springer, Netherlands, Vol. 55, pp. 345 - 348, 1992.

Yu, J., Yip, L., & Makis, V. (2010). Wavelet analysis with time-synchronous averaging of planetary gearbox vibration data for fault detection, diagnostics, and condition based maintenance, *2nd International Conference on Mechanical and Electronics Engineering (ICMEE)*, Kyoto, Japan, Vol. 1, pp. 132 - 136.

Wu, B., Sa xena, A., Khawaja, T. S ., Patrick, R. , Vachtsevanos, G., & Sparis, P. (2004). An approach to fault diagnosis of helicopter planetary gears, *Proceedings of IEEE AUTOTESTCON*, San Antonio, TX., pp. 475 - 481.

BIOGRAPHIES

Jae Yoon received his B.E degree in Control engineering from Kwangwoon University, Republic of Korea, worked at Samsung Electronics Co. Ltd. as a product engineer from 2006 through 2008. He then recei ved M.S. degree in Mechanical engineering from the University of Florid a. He jo ined the Intelligent Systems Modeling & Development Laboratory in the department of Mechanical and Industrial Engineering at the University of Illinois-Chicago to pursue Ph.D. degree. His current research i nterests include: machinery health monitoring for CBM, data-driven methods for diagnostics, and model based and data mining based prognostics, encompassing reliability engineering.

David He Dr. He is a Profess or and Director of the Intelligent Systems Modeling & Development Laboratory in the Department of Mechanical and Industrial Engineering at The University of Illinois-Chicago. D r. He's research areas include: machinery health monitoring, diagnosis and pro gnosis, complex systems failure analysis, qu ality and reliab ility engineering, and manufacturing systems design, modeling, scheduling and planning.

Brandon Van Hecke received his B.S. in Industrial Engineering from the University of Illinois at Chicago in 2010. He is a Ph.D. candidate in the Department of Mechanical and Industrial Engineering. His research interests include digital signal processing, machinery health monitoring, bearing and gear fault diagnostics based on the evaluation of vibration and acoustic emission signals, and condition based maintenance.

Integrated Diagnostics and Prognostics for the Electrical Power System of a Planetary Rover

Matthew Daigle[1], Indranil Roychoudhury[2], and Anibal Bregon[3]

[1] *NASA Ames Research Center, Moffett Field, California, 94035, USA*
matthew.j.daigle@nasa.gov

[2] *SGT Inc., NASA Ames Research Center, Moffett Field, California, 94035, USA*
indranil.roychoudhury@nasa.gov

[3] *Department of Computer Science, University of Valladolid, Valladolid, Spain*
anibal@infor.uva.es

ABSTRACT

For electric vehicles, technology for monitoring, diagnosis, and prognosis of the electrical power system (EPS) becomes essential for safe and efficient operation. To this end, we develop a general system-level integrated diagnosis and prognosis framework, which detects, isolates, and identifies EPS faults, and predicts when the EPS will fail to deliver sufficient power. The approach takes advantage of recent work in structural model decomposition in order to distribute the global diagnosis and prognosis problems into local subproblems that can be solved in parallel, thus enabling implementation on distributed computational platforms. The framework is applied to the EPS of a planetary rover testbed, and is demonstrated using data from field experiments.

1. INTRODUCTION

For electric vehicles, technology for monitoring, diagnosis, and prognosis of the electrical power system (EPS) is critical. In order to ensure safety, algorithms are needed that are able to predict the end-of-discharge (EOD) of the batteries powering the vehicle. The EOD time depends both on the current state of the batteries, including state-of-charge (SOC), and the future power requirements of the batteries. The future power requirements for the batteries depend both on the power required for future vehicle maneuvers and on any fault present in the system, which may cause increases in power demands. Therefore, both diagnosis (determining the current system state and faults) and prognosis (predicting the EOD of the system) are required.

Matthew Daigle et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

A large body of research exists for both model-based diagnosis (Gertler, 1998; Blanke et al., 2006) and prognosis methods (Luo et al., 2008; Saha & Goebel, 2009; Orchard & Vachtsevanos, 2009), however, most of the approaches in the literature focus in either solely the diagnosis or the prognosis task. A few works have proposed the integration of both tasks within a common framework (Patrick et al., 2007; Orchard & Vachtsevanos, 2009; Roychoudhury & Daigle, 2011; Zabi et al., 2013), however, unlike our approach, these approaches perform the diagnosis and prognosis tasks in a centralized way, thus suffering from scalability issues due to the large number of states and parameters in real-world systems. Moreover, most solutions do not approach the system-level problem. To the best of our knowledge, there is no approach in the literature which combines, in a distributed fashion, the system-level diagnosis and prognosis tasks.

In previous work, we have developed an integrated model-based diagnosis and prognosis framework (Roychoudhury & Daigle, 2011). The main contribution of this work was a unified modeling framework. In an extension of this work, we used structural model decomposition to develop a distributed integrated diagnosis and prognosis framework (Bregon, Daigle, & Roychoudhury, 2012), based on other work in distributed diagnosis (Bregon et al., 2014) and distributed prognosis (Daigle, Bregon, & Roychoudhury, 2012, 2014). Through structural model decomposition, a global model is transformed into a set of local submodels. For model-based diagnosis and prognosis, this results in the global diagnosis and prognosis problems being transformed into local diagnosis and prognosis subproblems. These subproblems can be solved independently by assigning them to different processing units, thus enabling a scalable and computationally efficient distributed diagnosis and prognosis solution.

In this paper, we apply these frameworks and ideas to the EPS of a planetary rover testbed at NASA Ames Research Center (Balaban et al., 2013). The applied architecture constitutes a new framework for integrated *system-level* diagnosis and prognosis. For the rover, we are interested in a system-level prediction, that is, when the EPS can no longer supply sufficient power to the loads. The rover is powered by several batteries, and this condition is a function of the state of all the batteries. Hence, component-level prognostics algorithms cannot be used, and a system-level prognosis framework is required (Daigle, Bregon, & Roychoudhury, 2012). We utilize recent work in structural model decomposition (Roychoudhury, Daigle, Bregon, & Pulido, 2013) to achieve a distributed implementation of the framework. We demonstrate the complete approach using real experimental data from the rover operating in the field.

The paper is organized as follows. Section 2 formulates the system-level diagnosis and prognostics problems. Section 3 describes the background on structural model decomposition, distributed diagnosis, and distributed diagnosis. Section 4 presents the rover EPS case study. Sections 5 and 6 present the system-level diagnosis and prognostics solutions, respectively, for the rover EPS. Section 7 presents the results for different scenarios. Finally, Section 8 concludes the paper.

2. PROBLEM FORMULATION

In this section, we formulate the integrated system-level diagnosis and prognosis problem. Ultimately, the goal is to predict when some event occurs in the system, such as the rover running out of power. In order to make such a prediction, we need to know the state of the system, including any faults that are present, therefore, diagnosis is needed in order to perform prognosis. We first formulate the system-level diagnosis problem, followed by the system-level prognosis problem.

2.1. System-Level Diagnosis

The problem of system-level diagnosis consists of three parts: (*i*) detecting whether a fault is present, (*ii*) isolating the correct fault, and (*iii*) identifying the faulty system state. In each of these parts, different models may be used. We assume that a model $\mathcal{M}$ can be succinctly represented in the following general formulation:

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \quad (1)$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \quad (2)$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[1] We will describe in Section 3 an equivalent structural representation of a model $\mathcal{M}$ that will be used for structural model decomposition.

In the model-based paradigm, we assume that in the nominal (fault-free) case, the system behaves according to some model $\mathcal{M}_n$, and, given the inputs $\mathbf{u}(k)$, produces measured outputs $\mathbf{y}(k)$. The problem of fault detection is to determine when model-predicted (nominal) outputs $\hat{\mathbf{y}}_n(k)$ are different from the measured outputs $\mathbf{y}(k)$ in a statistically significant manner. The difference $\mathbf{y}(k) - \hat{\mathbf{y}}_n(k)$ is called a *residual*; a (statistically significant) nonzero residual indicates a fault.

Faults are generally represented as changes in the model (i.e., in parameter values and/or model structure). So, in general, each fault $f \in F$, where F is the complete set of potential faults, is represented as a new model, $\mathcal{M}_f$. Given that a fault is present, the problem of fault isolation is to determine which model $\mathcal{M}_f$ now represents the system. The problem of fault identification is to determine the fault parameter estimate for the isolated fault, $p(\boldsymbol{\theta}_f(k)|\mathbf{y}(k_0\!:\!k))$, where $\mathbf{y}(k_0\!:\!k)$ denotes all measurements observed from the initial time k_0 to the current time k.

2.2. System-Level Prognosis

Rather than being focused on individual components, system-level prognostics is focused on the system as a whole, and on predictions for the system. As such, it is a more general formulation of the prognostics problem. System-level prognostics was previously defined in (Daigle, Bregon, & Roychoudhury, 2012). Here, we generalize the problem formulation based on (Daigle & Kulkarni, 2014) and explicitly integrate it with the diagnosis problem. Specifically, predictions must be made for a given fault hypothesis, which consists of a fault model $\mathcal{M}_f$ and joint state-parameter estimate $p(\mathbf{x}_f(k), \boldsymbol{\theta}_f(k)|\mathbf{y}(k_0\!:\!k))$. Fault identification computes an estimate of $\boldsymbol{\theta}_f(k)$, and the initial step of prognostics is to compute the full joint-state parameter estimate for the new faulty model.

System-level prognostics is concerned with predicting the occurrence of some system-level event E that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold function $T_{E_f} : \mathbb{R}^{n_{x_f}} \times \mathbb{R}^{n_{\theta_f}} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$ changes from the value 0 to 1. That is, the time of the event k_{E_f} at some time of prediction k_P given some fault f is defined as

$$k_{E_f}(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_{E_f}(\mathbf{x}_f(k), \boldsymbol{\theta}_f(k), \mathbf{u}(k)) = 1\}. \quad (3)$$

[1] Bold typeface denotes vectors, and n_a denotes the length of a vector $\mathbf{a}$.

The time remaining until that event, Δk_{E_f}, is defined as

$$\Delta k_{E_f}(k_P) \triangleq k_{E_f}(k_P) - k_P. \quad (4)$$

The prognostics problem is inherently uncertain, due to the random nature of the system evolution (represented with $\mathbf{v}(k)$), and unknown future inputs ($\mathbf{u}(k)$ for $k > k_P$). Therefore, k_{E_f} and Δk_{E_f} are random variables, and we must compute the probability distribution $p(k_{E_f}(k_P)|\mathbf{y}(k_0{:}k_P))$ (Daigle, Saxena, & Goebel, 2012; Sankararaman, Daigle, Saxena, & Goebel, 2013; Sankararaman, Daigle, & Goebel, 2014).

3. BACKGROUND

For a large system, both the diagnosis and prognosis problems are correspondingly large. A centralized approach does not scale well, can be computationally expensive, and prone to single points of failure. Therefore, we propose to decompose the *global* diagnosis and prognosis problems into independent *local* subproblems. In this work, we build on the ideas from structural model decomposition (Blanke et al., 2006; Pulido & Alonso-González, 2004) to compute local independent subproblems, which may be solved in parallel, thus providing scalability and efficiency.

We adopt here the structural model decomposition framework described in (Roychoudhury et al., 2013). This approach allows us to make guarantees of the minimality of the derived submodels and allows to generate different submodels for each one of the diagnosis and prognosis tasks. In the following, we review the main details and refer the reader to (Roychoudhury et al., 2013) for additional explanation. We define a model as follows:

Definition 1 (Model)**.** A *model* $\mathcal{M}^*$ is a tuple $\mathcal{M}^* = (V, C)$, where V is a set of variables, and C is a set of constraints among variables in V. V consists of five disjoint sets, namely, the set of state variables, X; the set of parameters, Θ; the set of inputs, U; the set of outputs, Y; and the set of auxiliary variables, A. Each constraint $c = (\varepsilon_c, V_c)$, such that $c \in C$, consists of an equation ε_c involving variables $V_c \subseteq V$.

Input variables, U, are known, and the set of output variables, Y, correspond to the (measured) sensor signals. Parameters, Θ, include explicit model parameters that are used in the model constraints. Auxiliary variables, A, are additional variables that are algebraically related to the state and parameter variables, and are used to reduce the structural complexity of the equations.

The notion of a *causal assignment* is used to specify the computational causality for a constraint c, by defining which $v \in V_c$ is the dependent variable in equation ε_c.

Definition 2 (Causal Assignment)**.** A *causal assignment* α to a constraint $c = (\varepsilon_c, V_c)$ is a tuple $\alpha = (c, v_c^{out})$, where $v_c^{out} \in V_c$ is assigned as the dependent variable in ε_c.

We write a causal assignment of a constraint using its equation in a causal form, with $:=$ to explicitly denote the causal (i.e., computational) direction.

Definition 3 (Valid Causal Assignments)**.** We say that a set of causal assignments $\mathcal{A}$, for a model $\mathcal{M}^*$ is *valid* if

- For all $v \in U \cup \Theta$, $\mathcal{A}$ does not contain any α such that $\alpha = (c, v)$.
- For all $v \in Y$, $\mathcal{A}$ does not contain any $\alpha = (c, v_c^{out})$ where $v \in V_c - \{v_c^{out}\}$.
- For all $v \in V - U - \Theta$, $\mathcal{A}$ contains exactly one $\alpha = (c, v)$.

The definition of valid causal assignments states that (*i*) input or parameter variables cannot be the dependent variables in the causal assignment, (*ii*) a measured variable cannot be used as an independent variable in any constraint, and (*iii*) every variable, which is not input or parameter, is computed by only one (causal) constraint.

Based on this, a *causal model* is a model extended with a valid set of causal assignments.

Definition 4 (Causal Model)**.** Given a model $\mathcal{M}^* = (V, C)$, a *causal model* for $\mathcal{M}^*$ is a tuple $\mathcal{M} = (V, C, \mathcal{A})$, where $\mathcal{A}$ is a set of valid causal assignments.

3.1. Structural Model Decomposition

To decompose a model into submodels, we need to break internal variable dependencies. We do this by selecting certain variables as local inputs. Given the set of potential local inputs (in general, selected from V), and the set of variables to be computed by the submodel (selected from $V - U - \Theta$), we create from a causal model $\mathcal{M}$ a causal submodel $\mathcal{M}_i$, in which a subset of the variables in V are computed using a subset of the constraints in C. In this way, each submodel computes independently from all other submodels. A causal submodel can be defined as follows.

Definition 5 (Causal Submodel)**.** A *causal submodel* $\mathcal{M}_i$ of a causal model $\mathcal{M} = (V, C, \mathcal{A})$ is a tuple $\mathcal{M}_i = (V_i, C_i, \mathcal{A}_i)$, where $V_i \subseteq V$, $C_i \subseteq C$, and $\mathcal{A}_i \cap \mathcal{A} \neq \varnothing$.

When using measurements (from Y) as local inputs, the causality of these constraints must be reversed, and so, in general, $\mathcal{A}_i$ is not a subset of $\mathcal{A}$.

The procedure for generating a submodel from a causal model is given as Algorithm 1 (`GenerateSubmodel`) in (Roychoudhury et al., 2013). Given a causal model $\mathcal{M}$, a set of variables that are considered as local inputs, U^*, and a set of variables to be computed, V^*, the `GenerateSubmodel` algorithm derives a causal submodel $\mathcal{M}_i$ that computes V^* using U^*. The algorithm works by starting at the variables in V^*, and propagating backwards through the causal dependencies. Propagation along a dependency chain stops once a variable in U^* is reached, or once a constraint is reached in which the causality can be reversed so that a variable in U^* can become a local input. We refer

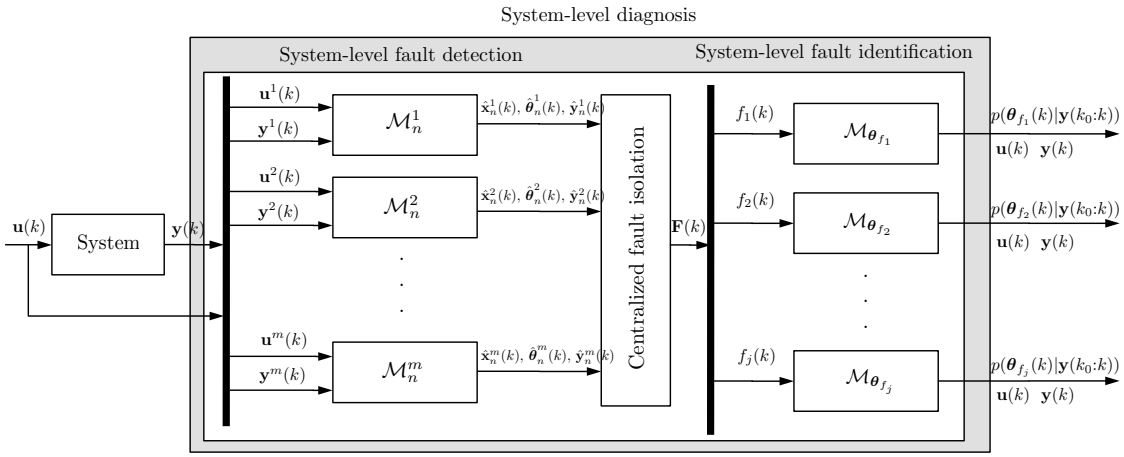


Figure 1. System-level diagnosis architecture.

the reader to (Roychoudhury et al., 2013) for the algorithm and additional details.

3.1.1. Structural Model Decomposition for System-Level Diagnosis

In this work, we use model decomposition to simplify the fault detection and fault identification problems (Bregon, Biswas, & Pulido, 2012; Bregon, Daigle, & Roychoudhury, 2012). For fault detection, we compute a set of residuals based on the sensors, and so derive a set of minimal local submodels to compute the nominal values of these sensors, i.e., one submodel for each $y \in Y$. In the submodel computing the output y, we use the other sensors $Y - \{y\}$ as local inputs, thus allowing decomposition. So, given the nominal model $\mathcal{M}_n$, for each output $y \in Y$, we create a submodel with $V^* = \{y\}$ and $U^* = \{U \cup (Y - \{y\})\}$.

Fault identification requires estimating a set of parameters associated with faults. Here, we also add Y as local inputs. Given a fault model $\mathcal{M}_f$, we create a submodel with $V^* = \boldsymbol{\theta}_f$, where $\boldsymbol{\theta}_f$ denotes the set of fault parameters, and $U^* = U \cup Y$.

3.1.2. Structural Model Decomposition for System-Level Prognosis

Prediction requires determining k_{E_f} for a given fault hypothesis f, which is computed based on T_{E_f}, which, in turn, is a function of the system states, parameters, and inputs. Often, the system-level, global threshold T_{E_f} can be expressed as the logical or of other local thresholds, i.e., $T_{E_f} = T_{E_f^1} \vee T_{E_f^2} \vee \ldots \vee T_{E_f^n}$ for n conditions. With each local threshold $T_{E_f^i}$ we can associate a local event E_f^i and compute times $k_{E_f^i}$, such that k_{E_f} can now also be defined as $min(k_{E_f^1}, k_{E_f^2}, \ldots, k_{E_f^n})$. This leads to a natural decomposition where each $k_{E_f^i}$ is computed independently, and allows us to decompose the prediction problem. So, to create the prediction submodels, we use the `GenerateSubmodel` algorithm in (Roychoudhury et al., 2013) with U^* set to $\{U_P\}$ and V^* set to $\{k_{E_f^i}\}$ for each local threshold $T_{E_f^i}$, where $U_P \subseteq V$ is the set of variables that can be predicted a priori.

The decomposition that can be achieved depends also on the selected U_P. If no variables exist that can be predicted a priori outside of U, then the `GenerateSubmodel` algorithm may not result in any decomposition and it will suffice to simply use the global model.

The initial state needed for prediction can be generated from a set of local estimators. The global prediction model is decomposed into local state estimators for the needed states, in the same way as in estimation for diagnosis.

3.2. Integrated System-level Diagnosis and Prognostics Architecture

Figs. 1 and 2 illustrate the architecture for our system-level diagnosis and prognosis frameworks, respectively. Regarding system-level diagnosis (Fig. 1), at each discrete time step, k, the system takes as input $\mathbf{u}(k)$ and produces outputs $\mathbf{y}(k)$. These are split into local inputs $\mathbf{u}^i(k)$ and local outputs $\mathbf{y}^i(k)$ for each one of the m system-level fault detection submodels, $\mathcal{M}_n^i$. Within each submodel $\mathcal{M}_n^i$, nominal tracking is performed, computing estimates of nominal states, $\hat{\mathbf{x}}_n^i(k)$, parameters, $\hat{\boldsymbol{\theta}}_n^i(k)$, and the measurements, $\hat{\mathbf{y}}_n^i(k)$. The fault isolator performs detection first by comparing the estimated measurement values against the observed values, to determine statistically significant deviations for the residual, $\mathbf{r}^i(k) = \mathbf{y}^i(k) - \hat{\mathbf{y}}^i(k)$. Deviations in the residuals are then transformed to qualitative symbols used by the centralized fault isolation block to generate a set of isolated fault candidates, $\mathbf{F}(k)$. For each one of the isolated fault candidates, $f_i(k)$, local models for fault identification, $\mathcal{M}_{\boldsymbol{\theta}_{f_i}}$, are used to compute local parameter estimates $p(\boldsymbol{\theta}_{f_i}(k)|\mathbf{y}(k_0{:}k))$. These

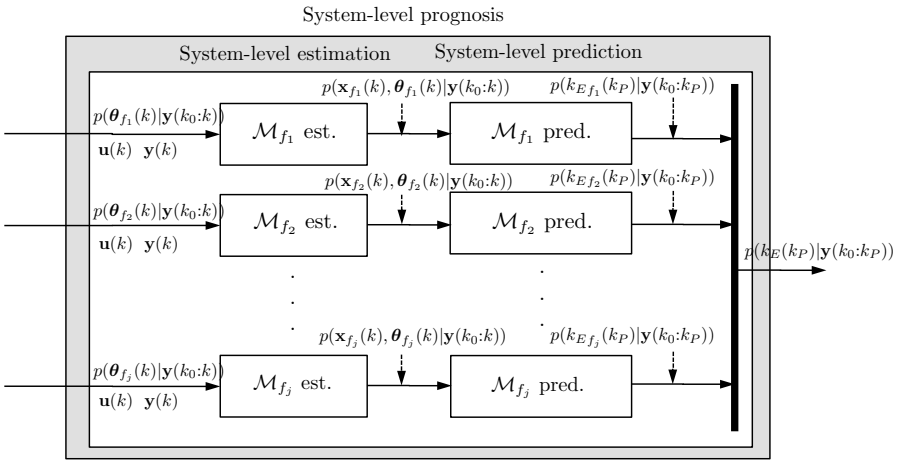


Figure 2. System-level prognosis architecture.

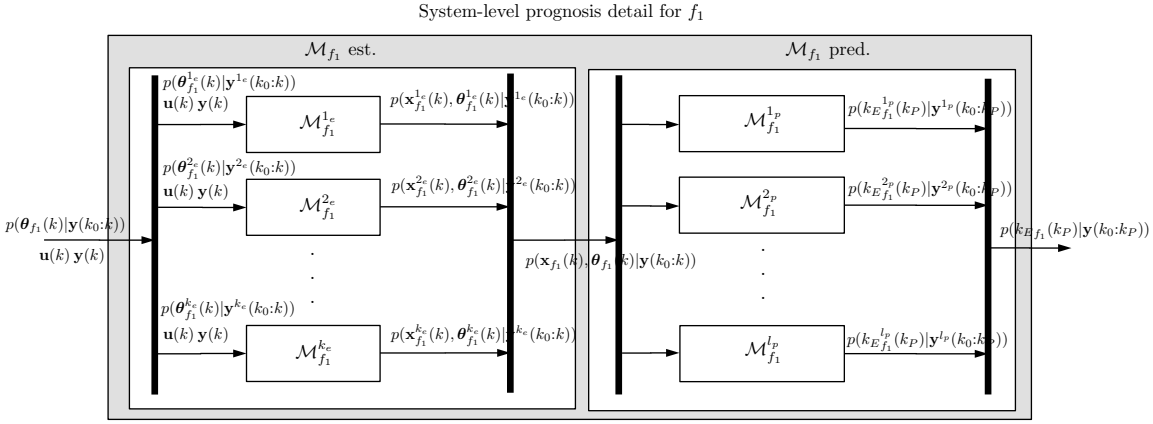


Figure 3. Detail of the system-level prognosis architecture.

local parameter estimates are then used as input to system-level prognosis (Fig. 2).

The system-level prognosis block of the architecture is divided into two phases: system-level estimation and system-level prediction. Parameter estimates from the local fault identification blocks, together with the inputs and outputs of the system, are used as input for the local estimation blocks, $\mathcal{M}_{f_i}$ *est.*, to compute state-parameter estimates $p(\mathbf{x}_{f_i}(k), \boldsymbol{\theta}_{f_i}(k)|\mathbf{y}(k_0{:}k))$. Finally, the local state-parameter estimates are used as input to the system-level prediction blocks, $\mathcal{M}_{f_i}$ *pred.*, to compute predictions, $p(k_{E_{f_i}}(k_P)|\mathbf{y}(k_0{:}k_P))$, at given prediction time k_P. Predictions for each fault hypothesis are combined into the global prediction $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$.

Fig. 3 shows the detail of the system-level estimation and prediction blocks for fault f_1, namely $\mathcal{M}_{f_1}$ *est.* and $\mathcal{M}_{f_1}$ *pred.* The system-level estimation task is decomposed using local estimation submodels, $\mathcal{M}^{1e}_{f_1}$ to $\mathcal{M}^{ke}_{f_1}$. As shown in the figure, subsets of the the local parameter estimates $p(\boldsymbol{\theta}_{f_1}(k)|\mathbf{y}(k_0{:}k))$, the system inputs, $\mathbf{u}(k)$, and the system outputs, $\mathbf{y}(k)$, are used as input for each one of the local state-parameter estimation submodels (this, of course, is similar to the estimation problem using the nominal model in the diagnosis part). The output of all the local submodels is then combined to compute the local state-parameter estimate for fault f_1, $p(\mathbf{x}_{f_1}(k), \boldsymbol{\theta}_{f_1}(k)|\mathbf{y}(k_0{:}k))$. The system-level prediction problem is also decomposed using local prediction submodels. The state estimate for the fault is split into local estimates for the prediction submodels, which then each compute a local $k^i_{E_{f_1}}$ value; these are then merged into the system-level prediction $k_{E_{f_1}}$ for the fault.

4. ROVER EPS MODELING

We are interested in integrated diagnosis and prognosis of the EPS of the rover. Thus, our system under consideration consists of the batteries, the battery current sensor, and the voltage sensors. The rover motors, which produce the electrical loads experienced by the EPS, are considered outside of our system under consideration, and so the loads the motors demand are viewed as inputs to the EPS.

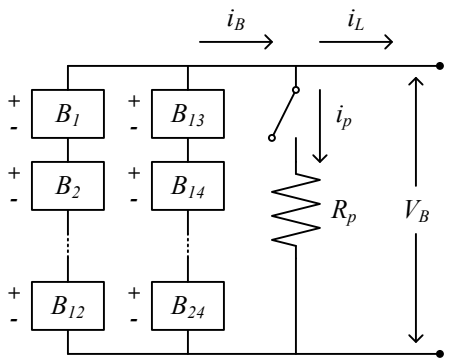


Figure 4. Rover EPS schematic.

The circuit schematic for the rover EPS is shown as Fig. 4. There are 24 lithium-ion cells in total, with two parallel branches of 12 cells in series. In parallel is a parasitic load, modeled as a resistance, R_p, that may appear as a fault. The battery current, i_B, is split into the current going to the load, i_L, and the current going to the parasitic load (if present), i_p. The total voltage provided by the EPS to the load is denoted as V_B. The cell model computes the voltage as a function of time given the current drawn from the cell, and is described in detail in (Daigle & Kulkarni, 2013). For completeness, the model is summarized in the appendix, and we refer the reader to (Daigle & Kulkarni, 2013) for additional explanation.

We assume that all cells start fully charged, so the voltage over each parallel branch is the same, and the current is split evenly ($i_B/2$). As the cells discharge, the total voltages must stay balanced, since the two sets of cells are in parallel, and therefore the current into each branch remains $i_B/2$.

The causal graph corresponding to the EPS model is shown in Fig. 5. The boxes in the figure indicate the battery cell models (for brevity, the internal variables are not shown). Also indicated are the sensor models. A measured value y^* (the * superscript indicates the measured value of a physical variable y) is equal to the physical variable y plus a bias, indicated with the ^{b} superscript. The biases, when present, produce a constant offset to the true value. Here, it also makes clear that we use the measured value of the load current, i_L^*, as an input to the system, which we assume is faultless.

The causal graph also indicates the computation of the time k_E (in the following, and in the figures, we drop the f subscript, as these submodels are not specific to a given fault). For the rover, E corresponds to any of the batteries reaching end-of-discharge (EOD), which is what must be predicted. EOD is defined by a voltage threshold V_{EOD}, where T_E is defined by $V_1 < V_{EOD}$ or $V_2 < V_{EOD}$, ..., $V_{24} < V_{EOD}$. When any cell voltage is less than V_{EOD}, EOD is reached for that battery and T_E evaluates to 1. The rover cannot be used beyond that point, as it will damage any batteries whose voltage is below the cutoff voltage.

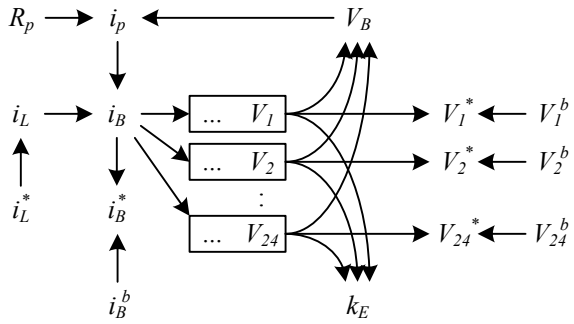


Figure 5. Causal graph for rover EPS model.

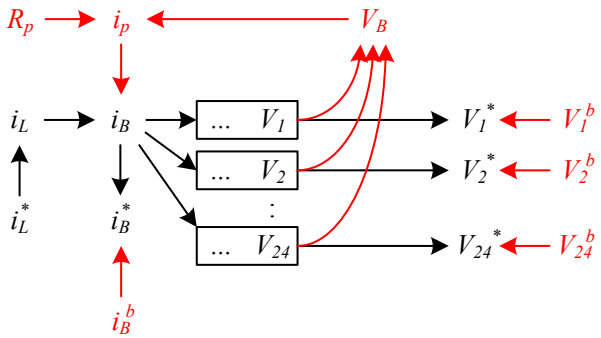


Figure 6. Causal graph for global nominal model.

5. ROVER EPS DIAGNOSIS

As described in Section 2, for diagnosis, models are used for the three phases of the diagnosis process: (*i*) fault detection, consisting of state estimation and residual generation, (*ii*) fault isolation, and (*iii*) fault identification. We describe the models used for each in the following subsections.

5.1. Fault Detection

Recall that in order to detect faults, we produce residuals, for which we need to compute model-predicted values of the outputs. We denote a residual using r_{y^*}, where y^* is the variable name for the sensor output. The causal graph for the global model for residual generation is shown as Fig. 6. It is generated by calling `GenerateSubmodel` with $U^* = \{i_L^*\}$, and $V^* = \{V_1^*, V_2^*, \ldots, V_{24}^*, i_B^*\}$. For residual generation, only the nominal model is needed, because the aim is only to detect when the nominal model is no longer valid, due to the appearance of a fault. In Fig. 6, the nominal parts of the model are colored black, and the fault-related parts in red. Since the faults are free from the nominal version of the model, only the black portion is needed for residual generation. We retain the red parts in the figures to indicate that the measured values will be causally effected by the faults.

As described in Section 3, we can decompose the residual generation problem, by creating local models for each sen-

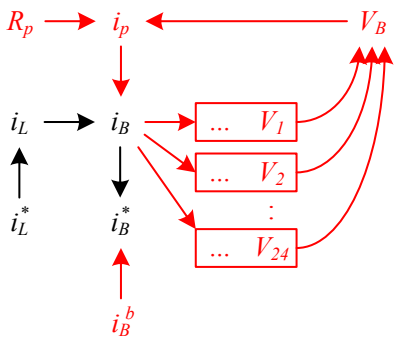


Figure 7. Causal graph for local i_B^* residual generator.

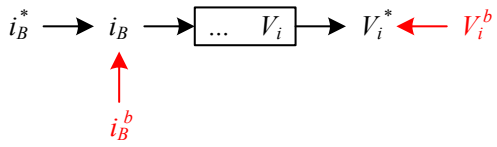


Figure 8. Causal graph for local V_i^* residual generator.

sor to compute predicted values. The causal graph for the local model for i_B^* is shown in Fig. 7, and is generated by calling `GenerateSubmodel` on the global model with $U^* = \{i_L^*, V_1^*, V_2^*, \ldots, V_{24}^*\}$ and $V^* = \{i_B^*\}$. The predicted value of i_B^*, in the nominal case, is simply equal to the measured load current, i_L^*. For each residual generator that has states we use the unscented Kalman filter (UKF) for estimation (Julier & Uhlmann, 2004).

The causal graph for the local model for V_i^* ($i \in [1, 24]$) is shown in Fig. 8, and is generated by calling `GenerateSubmodel` on the global model with $U^* = \{i_L^*, i_B^*, V_1^*, V_2^*, \ldots, V_{24}^*\} - \{V_i^*\}$ and $V^* = \{V_i^*\}$. The voltage for each cell is computed independently, using i_B^* as an input (this is divided by 2 to be used as input to the cell model).

5.2. Fault Isolation

Fault isolation is performed by analysis of the residual signals. Due to the decomposition used in the residual generation step, each fault manifests in only a subset of the complete residual set. As is clear in Fig. 7, $r_{i_B^*}$ will deviate (in a statistically significant way from zero) due only to the R_p fault and the i_B^b fault. As is clear in Fig. 8, $r_{V_i^*}$ will deviate due to V_i^b and i_B^b. Note that the relation $i_B^* = i_B + i_B^b$ holds, so when i_B^* is used as a local input, the causal relation is modified so that i_B becomes the dependent variable, and the causal constraint is $i_B := i_B^* - i_B^b$. That is, the true value of i_B is equal to the measured value minus the bias. For residual generation, the bias is not included, so by using the measured value, i_B^* as a local input, when a bias is present the wrong (i.e., biased) current will be fed to the cell model and used to compute V_i^*, thus causing a deviation in the corresponding residual.

The effects of the faults on the residuals are shown in Table 1. Faults are indicated both by the model parameter and the direction of its change, e.g., R_p^- denotes a decrease in the parasitic resistance.[2] Fault effects on residuals are represented as qualitative fault signatures (Mosterman & Biswas, 1999) and relative residual orderings (Daigle, Koutsoukos, & Biswas, 2007). Fault signatures express the qualitative change in a signal as the result of a fault. In general, they can be used to represent changes in magnitude, slope, and higher-order derivatives of a signal, but here, we represent changes in magnitude only, as this is sufficient to obtain unique diagnoses. For example, the parasitic load fault causes an increase in $r_{i_B^*}$. An ordering between a residual r_1 and r_2 for fault f, denoted as $r_1 \prec_f r_2$, indicates that the fault will cause an observable deviation in r_1 before r_2. For example, a bias in the V_1^* sensor will produce a deviation in $r_{V_1^*}$ before every other residual (since the fault affets no other residuals). Both signatures and orderings can be derived from the model automatically (Daigle, 2008).

Both signatures and orderings are reasoned over in an event-based framework to perform fault isolation (Daigle, Koutsoukos, & Biswas, 2009). When a residual deviation is first detected, the fault isolation algorithm checks for the faults that could have produced that deviation. As more residuals deviate, the algorithm checks for consistency with the current sequence of deviations, retaining only faults that can produce the observed sequence according to the predicted signatures and orderings. In addition, we can also eliminate candidates as inconsistent when no deviation is observed in a residual by using timeouts (this is equivalent to "observing" a `0` signature) (Daigle, Roychoudhury, & Bregon, 2013). For each residual we set a time limit under which we expect a residual deviation to occur after a fault. If we detect a fault and that residual has not deviated by that time, we observe a `0` signature and reason with that information. Including this information, we can distinguish qualitatively between all faults, and therefore obtain unique diagnoses based on the qualitative signatures and orderings alone.[3]

5.3. Fault Identification

The fault identification submodels are generated from the global model shown in Fig. 5, with the faulty parts included. In the call to `GenerateSubmodel`, U^* is set to the set of measured variables, and V^* is set to the fault parameter that is to be estimated.

[2] In the nominal model, when the parasitic load is absent, this is equivalent to an infinite resistance in parallel. Thus, the appearance of the parasitic load is denoted as a *decrease* in the parasitic resistance.

[3] Without using the `0` signatures for isolation, if a a voltage sensor bias occurred, we would have to wait infinitely long to ensure there were no further deviations and rule out i_B^b as a possibility.

Table 1. Fault Signatures and Residual Orderings

Fault	$r_{V_1^*}$	$r_{V_2^*}$	...	$r_{V_{24}^*}$	$r_{i_B^*}$	Residual Orderings
R_p^-	0	0	...	0	+	$r_{i_B^*} \prec r_{V_1^*}, r_{i_B^*} \prec r_{V_2^*}, \ldots, r_{i_B^*} \prec r_{V_{24}^*}$
V_1^{b+}	+	0	...	0	0	$r_{V_1^*} \prec r_{V_2^*}, \ldots, r_{V_1^*} \prec r_{V_{24}^*}, r_{V_1^*} \prec r_{i_B^*}$
V_1^{b-}	−	0	...	0	0	$r_{V_1^*} \prec r_{V_2^*}, \ldots, r_{V_1^*} \prec r_{V_{24}^*}, r_{V_1^*} \prec r_{i_B^*}$
V_2^{b+}	0	+	...	0	0	$r_{V_2^*} \prec r_{V_1^*}, \ldots, r_{V_2^*} \prec r_{V_{24}^*}, r_{V_2^*} \prec r_{i_B^*}$
V_2^{b-}	0	−	...	0	0	$r_{V_2^*} \prec r_{V_1^*}, \ldots, r_{V_2^*} \prec r_{V_{24}^*}, r_{V_2^*} \prec r_{i_B^*}$
...	...	...	...	...	...	...
V_{24}^{b+}	0	0	...	+	0	$r_{V_{24}^*} \prec r_{V_1^*}, \ldots, r_{V_{24}^*} \prec r_{V_2^*}, r_{V_{24}^*} \prec r_{i_B^*}$
V_{24}^{b+}	0	0	...	−	0	$r_{V_{24}^*} \prec r_{V_1^*}, \ldots, r_{V_{24}^*} \prec r_{V_2^*}, r_{V_{24}^*} \prec r_{i_B^*}$
i_B^{b+}	+	+	...	+	+	$\varnothing$
i_B^{b-}	−	−	...	−	−	$\varnothing$

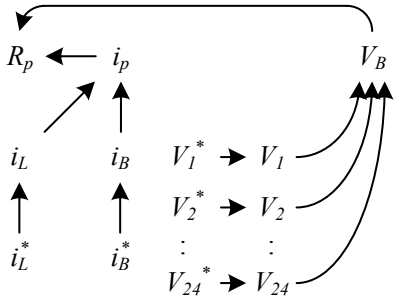


Figure 9. Causal graph for local R_p estimation.

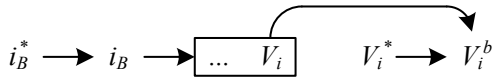


Figure 10. Causal graph for local V_i^b estimation.

For the parasitic load fault, the causal graph for the local estimation model is shown in Fig. 9. The parasitic resistance R_p is computed using i_P and V_B, where i_p is computed based on the difference between the measured load and battery currents, and V_B is computed based on the measured voltages.

The causal graph for the local model for the voltage sensor bias estimation is shown in Fig. 10. The voltage bias is computed based on the measured voltage and the model-predicted voltage, computed using the measured battery current.

The causal graph for the local model for the current sensor bias estimation is shown in Fig. 11. i_B^b is computed as the difference between the measured battery and load currents.

6. ROVER EPS PROGNOSIS

As described in Section 2, prognosis requires a prediction model, an initial state estimate, and future trajectories of the inputs, $\mathbf{U}_{k_P}$ and the process noise, $\mathbf{V}_{k_P}$. The prediction model must be able to compute the event threshold T_E, given the local inputs for prediction.

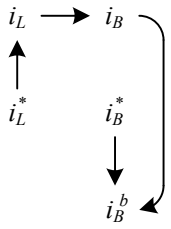


Figure 11. Causal graph for local i_B^b estimation.

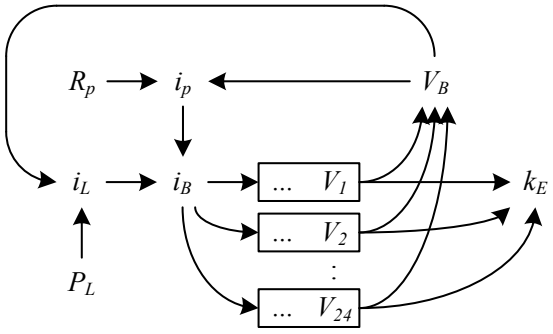


Figure 12. Causal graph for system-level prediction.

The causal graph for the global model for prediction is shown in Fig. 12. We need only to compute T_E, so none of the sensor outputs are included. Note also that for prediction, we use as an input the load power, P_L, instead of the load current. This is because it is much easier in practice to predict load power a priori. With a given speed command to the rover motors, power is constant, but current will increase as the battery cells discharge and V_B decreases.

It is important also to note that the prediction problem cannot be decomposed in general. Given i_B, we can compute each V_i independently, and evaluate $V_i < V_{EOD}$. Since E occurs when any one of the cells drops below the cutoff voltage, we can compute EOD for each cell and take the minimum to determine when E will occur (since E occurs when the first cell reaches EOD). However, i_B depends on i_P and i_L, both

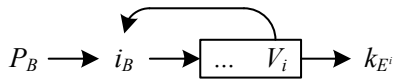


Figure 13. Causal graph for local prediction for cell i.

of which depend on V_B. There are no local inputs to break this dependency.

If we make a simplifying assumption, however, we can decompose the prediction problem and thus achieve the benefits of a distributed implementation. The causal graph for this case is shown in Fig. 13. In this case, we use as a local input the cell power P_B, where $P_B = P_L/24$, thus allowing local EOD thresholds, T_{E^i}, and, hence, local events E^i, to be computed independently. This assumption is only valid if the cells are all approximately equal in voltage, otherwise the assumption of $P_B = P_L/24$ will be violated. Further, this is not valid when R_p is present, as in that case P_B is a function of both P_L and i_p.

In general, the state estimates required for the prediction models must be produced by new estimators derived using structural model decomposition, for the global prediction model. For some faults, however, the needed estimates may be available from the residual generators, if those residual generators were not affected by the fault. In this case, new estimators do not need to be derived. For the parasitic load fault, the residual generator for each V_i^* has the state estimates for the battery cells, and the fault identifier has the value of R_p. We can then reconstruct a global state estimate for use in prediction. For a voltage sensor fault, a new local estimator (same as that used for residual generation, see Fig. 8) is needed to reestimate the states for the corresponding battery cell model. From the time of fault detection onwards, the corrected value of the sensor, computed by removing the estimated bias, is used to reestimate the states. For the current sensor fault, the case is more complex, because a faulty sensor reading was used in all of the local voltage estimators. Therefore, new local estimators are needed for all cells, in which the bias-corrected value must be fed as an input from the time of fault detection onward, once the fault bias has been identified.

In this work, we assume that process noise is negligible compared to the future input uncertainty, so represent the uncertainty only in the future input trajectories $\mathbf{U}_{k_P}$ (i.e, the trajectory of P_L). We use the surrogate variable method to represent the future input trajectories (Daigle & Sankararaman, 2013). In this method, we represent $\mathbf{U}_{k_P}$ through a set of surrogate variables, such that $\mathbf{U}_{k_P}$ can be constructed in a deterministic way given values of the surrogate variables. In this way, we can represent the probability distributions of the surrogate variables to indirectly represent the probability distribution of the input trajectories. For the rover, we consider an equivalent constant-loading distribution for the future inputs. That is, we assume that the future load power, P_L, will be constant with the value drawn from some distribution. In the case of the rover, the operator really only needs to know EOD predictions for best-, average-, and worst-case usage scenarios (Daigle & Kulkarni, 2014). For the state estimate, we use as samples the sigma points provided by the UKF. Each sample is simulated forward three times, once for each use case. From this we obtain best-, average-, and worst-case EOD predictions, each with some small variance (due to the state estimate variance).

It is important to note that since R_p is included in the prediction model, the prediction input does not change in the nominal and faulty cases. If, however, R_p was considered part of the load, i.e., part of P_L, then P_L prediction would have to change in the faulty case and would be complicated, since the additional power required by R_p is actually a function of battery voltage (as shown in Fig. 12). This is an advantage of viewing the prediction problem in a system-level perspective (the EPS perspective), rather than a component-level perspective (the battery cell perspective).

7. RESULTS

In this section, we demonstrate the integrated system-level diagnosis and prognosis framework on the rover case study, using real experimental field data. The task of the rover is to travel to different waypoints to complete some science objective. We must predict how long the rover will be able to execute its mission before having to return to the start point. Faults must be diagnosed so that the mission can be replanned if the rover is unable to meet all of its objectives due to the fault, and does not become stranded before returning to the start point.

We consider first a nominal scenario, in which the rover has enough energy to visit all waypoints and return successfully to the start point. Fig. 14 shows the measured and estimated values of V_1^* (results are similar for the remaining voltage sensors). With $V_{EOD} = 2.5$ V, EOD is clearly not reached. Fig. 15 shows tracking of the battery current sensor. Although the measured value is very noisy, the residual remains within the nominal range, and no fault is detected in any of the residuals. Fig. 16 shows the system-level EOD predictions for the rover. Each prediction consists of three points, for best-, average-, and worst-case future loading. Here, even in the worst-case scenario the predictions indicate that the rover will be able to complete the mission.

We next consider a parasitic load fault of 20 Ω, appearing as an additional load on the batteries, draining additional current and causing the batteries to discharge more quickly. The fault occurs at 780 s, and is detected at 801 s on the battery current residual, as shown in Fig. 17. Given the increase in the battery current, the parasitic load fault and a positive bias in

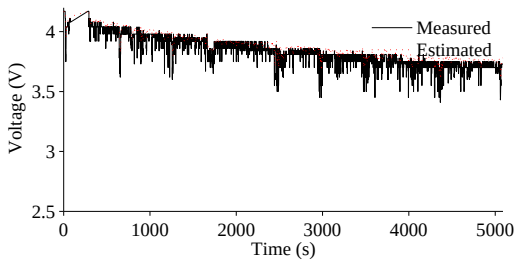


Figure 14. Estimation of V_1^*.

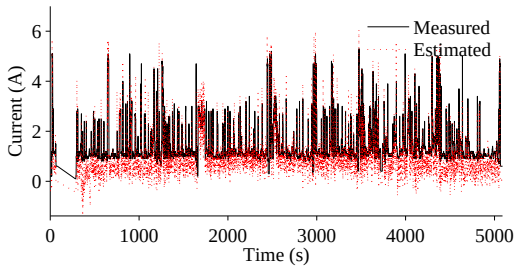


Figure 15. Estimation of i_B^*.

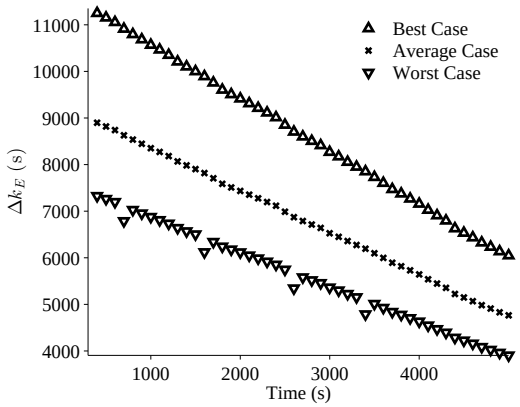


Figure 16. Predictions of Δk_E for worst-, average-, and best-case future usage scenarios.

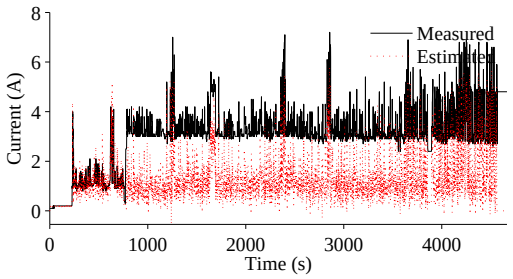


Figure 17. Estimation of i_B^* for a parasitic load.

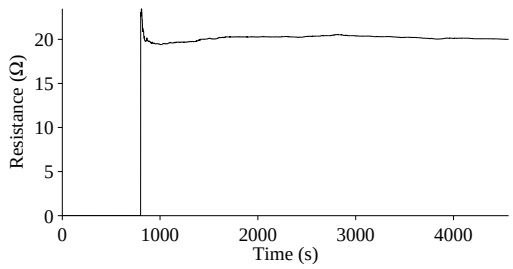


Figure 18. Estimation of R_p for a parasitic load.

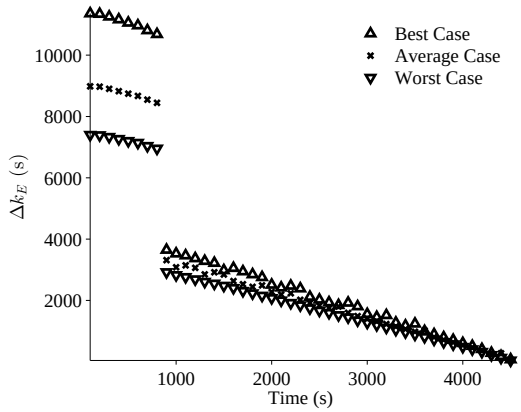


Figure 19. Predictions of Δk_E for worst-, average-, and best-case future usage scenarios with a parasitic load fault.

the battery current sensor are the only possible faults (see Table 1). At 922 s, two minutes after fault detection, we observe a 0 symbol on all the voltage sensor residuals, since they have not yet deviated. Given these observations, the only consistent candidate is the parasitic load fault. The estimated parasitic resistance over time is shown in Fig. 18. The estimate converges to the true value in less than 50 s, and stays very close to the true value. As described in Section 6, the prediction problem in this case cannot be decomposed, because the parasitic current depends on the battery voltages, so the local input for prediction is the total motor power. The system-level predictions are shown in Fig. 19. Before the fault is diagnosed, the predictions indicate that the rover will be able to complete its mission. After the fault is diagnosed, the predictions reflect the fact that more power is being demanded from the batteries, and EOD will be reached much sooner, requiring the mission to be shortened.

We next consider a battery voltage sensor fault, manifesting as a constant offset (bias) of 0.2 V on the voltage sensor for battery 1. The fault is injected at 600 s and detected at 634 s in the residual for the faulty sensor, as shown in Fig. 20. It is immediately diagnosed, as no other fault can produce a deviation first in the voltage sensor, according to the residual orderings. In order to recover from this fault, the estimator for the voltage is reset back to the estimated time of the fault, and is updated up to the current time using the unbiased signal, computed as the measured signal value minus the estimated bias. From the current time on, the present value of the estimated bias is used to correct the measured value sent to

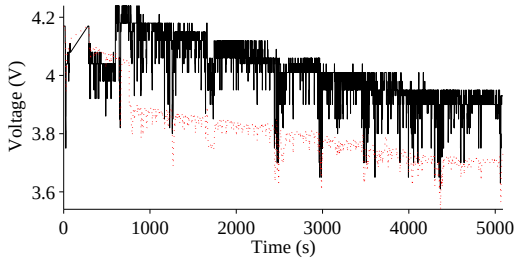


Figure 20. Estimation of V_1^* for the voltage sensor bias fault.

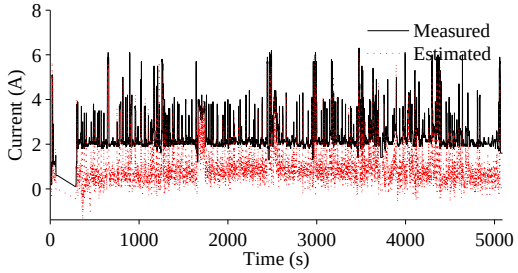


Figure 21. Estimation of i_B^* with a battery current sensor fault.

the estimator. Because this fault does not actually have any effect on the energy required by the rover, the predictions are the same as in the nominal condition.

Finally, we consider an offset fault in the battery current sensor. The fault is injected at 300 s, and is detected at 344 s. Detection time is slow due to the high amount of noise in the sensor. The tracking of the sensor is shown in Fig. 21, where the bias is clear visually (c.f. Fig. 15). The initial diagnosis is either the parasitic load fault, which can also cause an increase in the current, and a current sensor fault. Because a faulty current sensor value is being used as a local input to the voltage estimators, these residuals deviate as well. Tracking for V_1^* is shown in Fig. 22. Because a larger current is used, the estimated voltage drains faster than actual, and a deviation is detected at 415 s, thus isolating the current sensor fault as the true fault. Since the state estimates for the batteries will be corrupted, this will propagate to the predictions, giving incorrect results. So, to recover from the fault, once the fault is identified, the battery estimators are reset to the time of fault detection, and the corrected measurement value, based on the estimated bias is fed up to the current time and in the future. There is no physical effect on the energy consumption of the rover due to the fault, and therefore the predictions match those in the nominal case.

8. CONCLUSIONS

In this paper, we developed and implemented an approach for integrated system-level diagnosis and prognosis of the electrical power system of a planetary rover testbed. The algo-

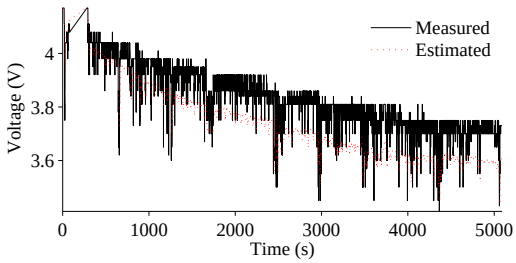


Figure 22. Estimation of V_1^* with a battery current sensor fault.

rithms monitor the behavior of the EPS and generate symbols for fault isolation in a distributed fashion. Fault isolation is performed, and for each fault hypothesis, system-level prognosis is performed, starting with distributed estimation of the state and fault parameters, and followed by distributed prediction. The distributed nature of the architecture is based upon the use of local submodels that enable the decomposition of global diagnosis and prognosis problems into local subproblems, applying ideas established in previous works. The approach was demonstrated using field data from the rover, showing successful detection, isolation, identification, and prediction for a set of realistic faults.

Future work will extend the application of the framework to the entire rover system, not just the EPS, which will enable the diagnosis of faults in the rover motors, and incorporation of that information into system-level predictions. We will also apply the approach to other systems, and make further theoretical extensions of the work, e.g., by including multiple faults, and hybrid systems.

ACKNOWLEDGEMENTS

The authors acknowledge Adam Sweet, NASA Ames Research Center, and George Gorospe, SGT, Inc., NASA Ames Research Center, for obtaining the experimental data used in this work. M. Daigle's and I. Roychoudhury's funding for this work was provided by the NASA System-wide Safety and Assurance Technologies (SSAT) Project. A. Bregon's funding for this work was provided by the Spanish MICINN DPI2013-45414-R grant.

REFERENCES

Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., & Gorospe, G. (2013). Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *Intl. Journal of Prognostics and Health Management*, *4*(1).

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and fault-tolerant control*. Springer.

Bregon, A., Biswas, G., & Pulido, B. (2012). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, *42*(3), 751-763.

Bregon, A., Daigle, M., & Roychoudhury, I. (2012, September). An integrated framework for model-based distributed diagnosis and prognosis. In *Annual conference of the prognostics and health management society 2012* (p. 416-426).

Bregon, A., Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., & Pulido, B. (2014, May). An event-based distributed diagnosis framework using structural model decomposition. *Artificial Intelligence*, *210*, 1-35.

Daigle, M. (2008). *A qualitative event-based approach to fault diagnosis of hybrid systems*. Unpublished doctoral dissertation, Vanderbilt University.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, September). A distributed approach to system-level prognostics. In *Annual conference of the prognostics and health management society 2012* (p. 71-82).

Daigle, M., Bregon, A., & Roychoudhury, I. (2014, June). Distributed prognostics based on structural model decomposition. *IEEE Trans. on Reliability*, *63*(2), 495-510.

Daigle, M., Koutsoukos, X., & Biswas, G. (2007, April). Distributed diagnosis in formations of mobile robots. *IEEE Trans. on Robotics*, *23*(2), 353–369.

Daigle, M., Koutsoukos, X., & Biswas, G. (2009, July). A qualitative event-based approach to continuous systems diagnosis. *IEEE Trans. on Control Systems Technology*, *17*(4), 780–793.

Daigle, M., & Kulkarni, C. (2013, October). Electrochemistry-based battery modeling for prognostics. In *Annual conference of the prognostics and health management society 2013* (p. 249-261).

Daigle, M., & Kulkarni, C. (2014). A battery health monitoring framework for planetary rovers. In *Proceedings of the ieee aerospace conference.*

Daigle, M., Roychoudhury, I., & Bregon, A. (2013, October). Qualitative event-based diagnosis with possible conflicts: Case study on the fourth intl. diagnostic competition. In *Proc. of the 24th intl. workshop on principles of diagnosis* (p. 230-235).

Daigle, M., & Sankararaman, S. (2013, October). Advanced methods for determining prediction uncertainty in model-based prognostics with application to planetary rovers. In *Annual conference of the prognostics and health management society 2013* (p. 262-274).

Daigle, M., Saxena, A., & Goebel, K. (2012, September). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society* (p. 326-335).

Gertler, J. J. (1998). *Fault detection and diagnosis in engineering systems*. New York, NY: Marcel Dekker, Inc.

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Karthikeyan, D. K., Sikha, G., & White, R. E. (2008). Thermodynamic model development for lithium intercalation electrodes. *Journal of Power Sources*, *185*(2), 1398–1407.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *29*(6), 554-565.

Orchard, M. E., & Vachtsevanos, G. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Trans. of the Institute of Measurement and Control*, *31*(3/4), 221-246.

Patrick, R., Orchard, M. E., Zhang, B., Koelemay, M., Kacprzynski, G., Ferri, A., & G., V. (2007, September). An integrated approach to helicopter planetary gear fault diagnosis and failure prognosis. In *Proc. of the 42nd annual systems readiness technology conf.* Baltimore, MD, USA.

Pulido, B., & Alonso-González, C. (2004). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, *34*(5), 2192-2206.

Rahn, C. D., & Wang, C.-Y. (2013). *Battery systems engineering*. Wiley.

Roychoudhury, I., & Daigle, M. (2011, October). An integrated model-based diagnostic and prognostic framework. In *Proc. of the 22nd intl. workshop on principles of diagnosis* (p. 44-51).

Roychoudhury, I., Daigle, M., Bregon, A., & Pulido, B. (2013, March). A structural model decomposition framework for systems health management. In *Proceedings of the 2013 ieee aerospace conference.*

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society 2009.*

Sankararaman, S., Daigle, M., & Goebel, K. (2014, June). Uncertainty quantification in remaining useful life prediction using first-order reliability methods. *IEEE Trans. on Reliability*, *63*(2), 603-619.

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013, March). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *Proc. of the 2013 IEEE aerospace conference.*

Zabi, S., Ribot, P., & Chanthery, E. (2013, October). Health

monitoring and prognosis of hybrid systems. In *Proc. of the annual conference of the prognostics and health management society 2013* (p. 300-311).

BIOGRAPHIES

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis of complex physical systems.

APPENDIX: BATTERY CELL MODELING

The battery cell model computes the voltage as a function of time given the current drawn from the cell, and is described in detail in (Daigle & Kulkarni, 2013). We summarize the model here and refer the reader to (Daigle & Kulkarni, 2013) for additional explanation.

The voltage terms of the battery are expressed as functions of the amount of charge in the electrodes (the states of the model). Each electrode, positive (subscript p) and negative (subscript n), is split into two volumes, a surface layer (subscript s) and a bulk layer (subscript b). The differential equations for the battery describe how charge moves through these volumes. The charge (q) variables are described using

$$\dot{q}_{s,p} = i_{app} + \dot{q}_{bs,p} \tag{5}$$
$$\dot{q}_{b,p} = -\dot{q}_{bs,p} + i_{app} - i_{app} \tag{6}$$
$$\dot{q}_{b,n} = -\dot{q}_{bs,n} + i_{app} - i_{app} \tag{7}$$
$$\dot{q}_{s,n} = -i_{app} + \dot{q}_{bs,n}, \tag{8}$$

where i_{app} is the applied electric current The term $\dot{q}_{bs,i}$ describes diffusion from the bulk to surface layer for electrode i:

$$\dot{q}_{bs,i} = \frac{1}{D}(c_{b,i} - c_{s,i}), \tag{9}$$

where D is the diffusion constant. The c terms are lithium ion concentrations:

$$c_{b,i} = \frac{q_{b,i}}{v_{b,i}} \tag{10}$$
$$c_{s,i} = \frac{q_{s,i}}{v_{s,i}}, \tag{11}$$

where, for CV v in electrode i, $c_{v,i}$ is the concentration and $v_{v,i}$ is the volume. We define $v_i = v_{b,i} + v_{s,i}$. Note now that the following relations hold:

$$q_p = q_{s,p} + q_{b,p} \tag{12}$$
$$q_n = q_{s,n} + q_{b,n} \tag{13}$$
$$q^{\max} = q_{s,p} + q_{b,p} + q_{s,n} + q_{b,n}. \tag{14}$$

We can also express mole fractions (x) based on the q vari-

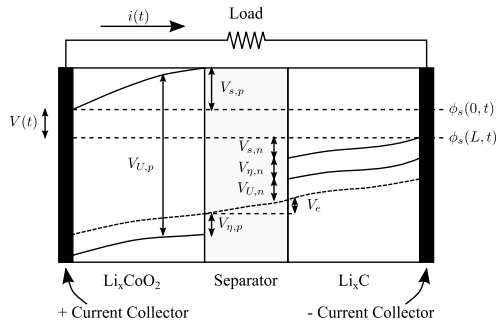


Figure 23. Battery voltages.

ables:

$$x_i = \frac{q_i}{q^{\max}}, \tag{15}$$

$$x_{s,i} = \frac{q_{s,i}}{q_{s,i}^{\max}}, \tag{16}$$

$$x_{b,i} = \frac{q_{b,i}}{q_{b,i}^{\max}}, \tag{17}$$

where $q^{\max} = q_p + q_n$ refers to the total amount of available Li ions. It follows that $x_p + x_n = 1$. For lithium ion batteries, when fully charged, $x_p = 0.4$ and $x_n = 0.6$. When fully discharged, $x_p = 1$ and $x_n = 0$ (Karthikeyan, Sikha, & White, 2008).

The different potentials are summarized in Fig. 23 (adapted from (Rahn & Wang, 2013)). The overall battery voltage $V(t)$ is the difference between the potential at the positive current collector, $\phi_s(0,t)$, and the negative current collector, $\phi_s(L,t)$, minus resistance losses at the current collectors (not shown in the diagram). At the positive current collector is the equilibrium potential $V_{U,p}$. This voltage is then reduced by $V_{s,p}$, due to the solid-phase ohmic resistance, and $V_{\eta,p}$, the surface overpotential. The electrolyte ohmic resistance then causes another drop V_e. At the negative electrode, there is a drop $V_{\eta,n}$ due to the surface overpotential, and a drop $V_{s,n}$ due to the solid-phase resistance. The voltage drops again due to the equilibrium potential at the negative current collector $V_{U,n}$. These voltages are described by the following set of equations (see (Daigle & Kulkarni, 2013) for details):

$$V_{U,i} = U_0 + \frac{RT}{nF} \ln\left(\frac{1 - x_{s,i}}{x_{s,i}}\right) + V_{\text{INT},i}, \tag{18}$$

$$V_{\text{INT},i} = \frac{1}{nF}\left(\sum_{k=0}^{N_i} A_{i,k}\left((2x_i - 1)^{k+1} - \frac{2x_i k(1 - x_i)}{(2x_i - 1)^{1-k}}\right)\right), \tag{19}$$

$$V_o = i_{app} R_o, \tag{20}$$

$$V_{\eta,i} = \frac{RT}{F\alpha}\text{arcsinh}\left(\frac{J_i}{2J_{i0}}\right), \tag{21}$$

$$J_i = \frac{i}{S_i}, \tag{22}$$

$$J_{i0} = k_i(1 - x_{s,i})^{\alpha}(x_{s,i})^{1-\alpha}, \tag{23}$$

$$V = V_{U,p} - V_{U,n} - V_o' - V_{\eta,p}' - V_{\eta,n}', \tag{24}$$

$$\dot{V}_o' = (V_o - V_o')/\tau_o \tag{25}$$

$$\dot{V}_{\eta,p}' = (V_{\eta,p} - V_{\eta,p}')/\tau_{\eta,p} \tag{26}$$

$$\dot{V}_{\eta,n}' = (V_{\eta,n} - V_{\eta,n}')/\tau_{\eta,n}. \tag{27}$$

Here, U_0 is a reference potential, R is the universal gas constant, T is the electrode temperature (in K), n is the number of electrons transferred in the reaction ($n = 1$ for Li-ion), F is Faraday's constant, J_i is the current density, and J_{i0} is the exchange current density, k_i is a lumped parameter of several constants including a rate coefficient, electrolyte concentration, and maximum ion concentration. $V_{\text{INT},i}$ is the activity correction term (0 in the ideal condition). We use the Redlich-Kister expansion with $N_p = 12$ and $N_n = 0$ (see (Daigle & Kulkarni, 2013)). The τ parameters are empirical time constants (used since the voltages do not change instantaneously).

The model contains as states $\mathbf{x}$, $q_{s,p}$, $q_{b,p}$, $q_{b,n}$, $q_{s,n}$, V_o', $V_{\eta,p}'$, and $V_{\eta,n}'$. The single model output is V.

The state of charge (SOC) of a battery is defined to be 1 when the battery is fully charged and 0 when the battery is fully discharged by convention. In this model, it is analogous to the mole fraction x_n, but scaled from 0 to 1. We distinguish here between nominal SOC and *apparent* SOC (Daigle & Kulkarni, 2013). Nominal SOC is computed based on the combination of the bulk and surface layer CVs in the negative electrode, whereas apparent SOC is be computed based only on the surface layer. When a battery reaches the voltage cutoff, apparent SOC is 0, and nominal SOC is greater than 0 (how much greater depends on the difference between the diffusion rate and the current drawn). Once the concentration gradient settles out, the surface layer will be partially replenished and apparent SOC will rise while nominal SOC remains the same. Nominal (n) and apparent (a) SOC are defined using

$$SOC_n = \frac{q_n}{0.6q^{\max}} \tag{28}$$

$$SOC_a = \frac{q_{s,n}}{0.6q^{\max_{s,n}}}, \tag{29}$$

where $q^{\max_{s,n}} = q^{\max}\frac{v_{s,n}}{v_n}$.[4]

[4]Note that SOC of 1 corresponds to the point where $q_n = 0.6q^{\max_{s,n}}$, since the mole fraction at the positive electrode cannot go below 0.4, as described earlier.

Demonstration of Prognostics-Enabled Decision Making Algorithms on a Hardware Mobile Robot Test Platform

Adam Sweet[1], George Gorospe[2], Matthew Daigle[1], José R. Celaya[2], Edward Balaban[1], Indranil Roychoudhury[2], and Sriram Narasimhan[3]

[1] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
{*adam.sweet, edward.balaban, matthew.j.daigle*}*@nasa.gov*

[2] *SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
{*george.e.gorospe, jose.r.celaya, indranil.roychoudhury*}*@nasa.gov*

[3] *University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035, USA*
sriram.narasimhan@nasa.gov

ABSTRACT

Prognostics-enabled Decision Making (PDM) is an emerging research area that aims to integrate prognostic health information and knowledge about the future operating conditions into the process of selecting subsequent actions for the system. Previous work developing and testing PDM algorithms has been done in simulation; this paper describes the effort leading to a successful demonstration of PDM algorithms on a hardware mobile robot platform. The hardware platform, based on the K11 planetary rover prototype, was modified to allow injection of selected fault modes related to the rover's electrical power subsystem. The PDM algorithms were adapted to the hardware platform, including development of a software module framework, a new route planner, and modifications to increase the algorithms' robustness to sensor noise and system timing issues. A set of test scenarios was chosen to demonstrate the algorithms' capabilities. The modifications to run with a hardware platform, the test scenarios, and the test results are described in detail. The results show a successful use of PDM algorithms on a hardware test platform to optimize mission planning in the presence of electrical system faults.

1. INTRODUCTION

The research fields of system health, diagnostics, and prognostics have become mature to the point where the techniques have begun to be incorporated in new designs of aerospace vehicles (Reveley, Kurtoglu, Leone, Briggs, & Withrow, 2010). This has led to the newer research area called Prognostics-enabled Decision Making (PDM), which is devoted to the ability to incorporate system health information in making decisions in the planning and control of the system. A vehicle capable of making decisions, or assisting a human operator to make decisions, based on system health information could potentially accomplish more mission objectives, or operate with improved safety margins, than those that do not incorporate those considerations.

A useful way to drive maturation of algorithms in diagnostics and prognostics has been to develop test platforms where the algorithms may be evaluated. NASA Ames Research Center has developed several such test platforms, first in the electrical power system domain (Poll et al., 2007) and in the electromechanical actuator domain (Smith et al., 2009; Balaban et al., 2010). Each test platform has provided a means for controlled injection of faults to test the capabilities of the diagnostic and prognostic algorithms and has driven their development to be robust to real-world issues such as data latency and sensor noise. However, each test platform was designed primarily with the diagnostic and prognostic problems in mind. This led to the development of another test platform - the mobile robot test platform for testing and maturation of PDM algorithms.

Work began on a mobile robot test platform (Balaban et al., 2011, 2013) to provide a means for maturing PDM algorithms and verifying their predictions in a real-world environment. As described in previous publications, the mobile robot test platform is expected to support the following high-level tasks: (*i*) development of system-level and component-level PDM algorithms; (*ii*) development of realistic fault injection and accelerated aging techniques for algorithm testing; (*iii*) maturation and standardization of interfaces between reasoning

Adam Sweet et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

algorithms; (*iv*) performance comparison of PDM algorithms from different sources; and (*v*) generation of publicly available datasets for enabling further PDM research. (Balaban et al., 2013) described the intended use of the test platform and the series of test scenarios which had been accomplished in simulation. This paper describes the adaptation of the algorithms to the hardware test platform, and the scenarios and results from using it to test PDM algorithms in the field.

The paper is organized as follows. Section 2 describes the platform modifications to support the new experiments. Section 3 presents the modifications to the PDM algorithms. Section 4 presents the experimental scenarios and results. Section 5 concludes the paper.

2. PLATFORM MODIFICATIONS

The ability to emulate realistic adverse events in the test platform is of key importance for the maturation process of PDM algorithms. In this context, an adverse event is regarded as an unexpected off-nominal physical change in the system under consideration. Such an event is to be properly observed by the health monitoring technology and properly mitigated or managed by the decisions and actions of the PDM system. Another important capability for a test platform is to provide a standard mechanism for its software modules to communicate with each other and with the PDM system. The adverse events emulated on the test platform and the software module framework are described in the sections below.

2.1. Hardware fault injection

The hardware faults currently implemented in the test platform are related to its electrical power system. As described in previous publications (Balaban et al., 2011, 2013), the rover vehicle under consideration is based on an electric power train in which the wheels are powered by electric motors and the power is stored in batteries. A variety of power conversion and mechanical faults in the electrical power train result in an increased power consumption in the form of higher levels of current demanded from the batteries. This ends up draining the batteries faster, thus potentially considerably reducing the duration of the rover mission. An example of an electrical power train fault that relates to increased energy consumption can be identified within the electrical motor controllers. A motor controller contains power switching elements like power transistors. The parasitic resistance of such devices and the lost of power dissipation capability due to degradation in performance during the device's lifetime, resulting in increased power consumption.

Because a variety of faults result in increased power consumption, the battery current drain circuit (parasitic load) was selected to implement on the robotic test platform. Other reasons for choosing that way of injecting hardware faults are that the circuit emulating the fault(s) has the ability to drain a variable amount of current and also that it is controlled programmatically. Figure 1 presents the battery current leakage circuit. It consists of three banks of resistors in parallel that can be engaged programmatically by closing the corresponding relay. The third bank in the diagram is a rheostat that is also controlled programmatically and ranges from 0 Ω to 10 Ω.

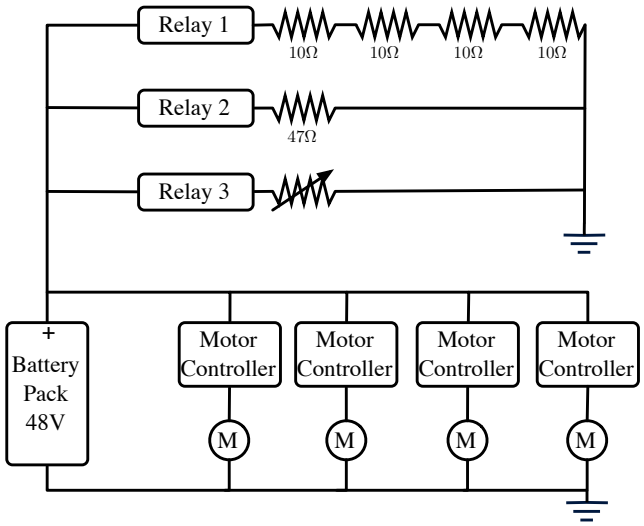


Figure 1. Battery current drain circuit schematic

2.2. Sensor fault injection

Sensor fault injection is another method of introducing faults in the mobile robot platform. The prognostics and decision making components of the PDM system depend on accurate knowledge of the platform's state, in order to make accurate predictions and correct decisions based on those predictions. If a sensor is faulty and results in an incorrect estimate of the system's state, it could lead to either suboptimal decisions or, in the aviation domain, the potential loss of the mobile robot platform. Therefore, injecting sensor faults on the mobile robot platform is a useful way to test the PDM system's robustness and ability to ensure correct decisions are being made even in the presence of these types of faults.

Common types of sensor faults were described in (Balaban, Saxena, Bansal, Goebel, & Curran, 2009; Poll et al., 2011), and, in the course of this work, three types of sensor faults were implemented: stuck, offset, and drift. When a sensor is stuck, its value is set to a specified value and is unchanging thereafter. When a sensor has an offset fault, its value differs from the correct value by some specified constant amount. Finally, when a sensor has a drift fault, its value diverges slowly from the correct value over time. Examples of these are shown graphically in Figure 2.

2.3. Software module framework

The hardware test platform requires software to operate. The software consists of three major subcomponents: (*i*) the rover control and data acquisition module; (*ii*) the reasoning algorithms, (*iii*) and the communication infrastructure. The rover control and data acquisition software is implemented in LabVIEW (*LabVIEW version 12.0.0.4029*, 2012) and is responsible for interacting with the rover hardware. Control of the rover is performed by specifying wheel speeds for each individual rover wheel through the wheel motor controller hardware. Data acquisition is performed by multiple devices, and the LabVIEW control software is responsible for gathering the data from all the devices and making it available as a single sensor array. This data is sent to the PDM system.

The communication infrastructure is responsible for facilitating information sharing between the rover control software and the reasoning algorithms, as well as among the various reasoning algorithm modules. This is accomplished through a publish/subscribe architecture, which is implemented through the Internet Communication Engine, ICE (Henning, 2004). Standardized interface definition files are used to describe messages exchanged among the software and hardware modules. The message types include command inputs, sensor data, vehicle state information, fault diagnosis candidates, as well as unordered and ordered waypoint lists. A central server coordinates message exchanges among any number of devices on the same network. In order to be integrated into the architecture, a new reasoning module needs to only implement a minimal interface to register with the ICE server and to publish and/or subscribe to the appropriate messages. For example, a diagnostic module would subscribe to the rover commands and sensor data and, in turn, publish diagnostic messages. Thus the architecture allows for easy accommodation of modules implemented in different programming languages and running on dissimilar platforms.

3. ALGORITHM MODIFICATIONS

Modifications were also made to several PDM system algorithms; namely the state of charge estimator, the electrical power system (EPS) diagnoser, the route planner, and the decision maker. The changes made to each are described below.

3.1. State-of-Charge Estimator

The battery state-of-charge (SOC) estimator employs a model-based approach. Whereas in (Balaban et al., 2013) an electric circuit equivalent model of the battery cell was used, in this work the underlying model employed is an electrochemistry model of the lithium-ion cell presented in (Daigle & Kulkarni, 2013). The model has higher accuracy, yet is based only on ordinary differential equations, and, like the equivalent circuit model, can be simulated very quickly, suitable for real-time operations. As in (Balaban et al., 2013), the unscented Kalman filter (UKF) is used for state estimation (Julier & Uhlmann, 2004). The UKF estimates internal model states, from which SOC and the cell voltage are computed.

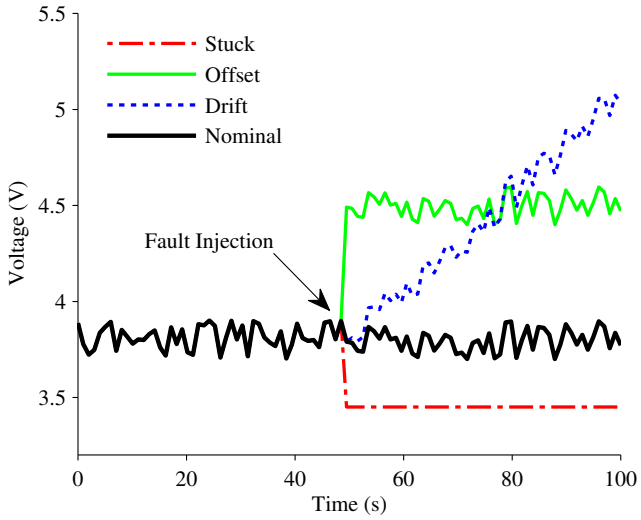


Figure 2. Examples of sensor faults

A distributed estimation approach (Daigle, Bregon, & Roychoudhury, 2014) can be used for the cells, where the local input to each cell's estimator is the measured battery current, i_B, divided by 2. Since the battery voltages on each parallel branch remain approximately balanced, the current going into the two branches is split evenly.

3.2. EPS Diagnoser

The diagnoser has three main diagnostic purposes, namely fault detection, isolation, and identification. Fault detection involves determining if a fault has occurred and is usually determined by taking the difference between the actual observed sensor readings and the model-predicted nominal behavior of these sensor readings, then determining if this difference is statistically significant. In order to compute the model-predicted signals, we adopt the structural model decomposition approach from (Roychoudhury, Daigle, Bregon, & Pulido, 2013) to decompose the global model of the EPS into smaller, local submodels, thus decomposing the model-based estimation problem. This is achieved by using measured sensor values as local inputs to the submodels. This, in fact, is what is done for the SOC estimators, formally justified by this decomposition approach. Thus, we obtain 25 local estimators, one for each cell voltage (using the SOC estimators), and one for the battery current (in which measured load current is used as an input, and we assume in the nominal case that the battery current is equal to the load current). The difference between a measured sensor value and the estimated value is termed the residual. A statistically significant

deviation of a residual from zero indicates a fault. The employed fault detection method, based on a Z-test, is detailed in (Daigle et al., 2010) and is the same as used in (Balaban et al., 2013).

Once a fault is detected, a qualitative event-based diagnosis (QED) approach (Daigle, Roychoudhury, & Bregon, 2013) is invoked for fault isolation. For each available residual, a symbol generation routine is invoked that transforms a quantitative residual into qualitative symbols $\{0, +, -\}$ indicating whether or not the observed sensor reading is at, above, or below the estimated nominal value, respectively. Fault isolation is performed by comparing these observed symbols with the model-predicted symbols (Mosterman & Biswas, 1999; Daigle, Koutsoukos, & Biswas, 2009). Fault candidates that are inconsistent with the observed sequence of residual deviations are dropped. The fault candidate set is continually pruned as more residual deviations are observed until, ideally, the true single fault is the only fault candidate remaining[1]. Since the residuals are computed using local submodels, most faults affect only a few residuals (e.g., the parasitic load causes a deviation only in the battery current residual). This is in contrast to the global estimation approach in (Balaban et al., 2013) in which faults affect many residuals. Using the distributed estimation approach improves diagnosability (Daigle, Bregon, Biswas, Koutsoukos, & Pulido, 2012).

Once the true fault is isolated, a local model for estimating the fault parameters is used. The state estimate is augmented with the fault estimate for use in the prediction and decision-making steps. In the case of sensor faults, the faulty sensor value may have been used as a local input to an estimator, thus corrupting the resulting estimates. Therefore, once the sensor fault is identified, the estimators that used that (faulty) sensor value as an input must be reset to the time of fault occurrence and run again up to the current time using the corrected sensor value, so that a correct state estimate (to be used for prediction) can be obtained.

3.3. Route Planner

The route planner is a new component responsible for determining the route that the test vehicle takes. It operates on a set of waypoints, which represent points of scientific interest. Each waypoint consists of a location, specified with latitude, longitude, and altitude, and a reward, which is an integer representing the scientific importance of that waypoint. In the case of an aerial vehicle or a high-level simulation, direct paths between waypoints can often be assumed. This is not generally the case for a ground vehicle, where terrain features and obstacles need to be taken into account when planning vehicle movement. The available waypoints are defined in advance and are located at the street intersections in the experiment's geographical area, as shown in Figure 3a. Not all of the defined waypoints were used as primary waypoints in the experiments described later; the choice of waypoints is described in Section 4. The waypoints which were used are shown as green in Figure 3a and the unused waypoints are shown in black. The waypoints are identified with numbers, shown in the figure just after the letter 'W' (for "waypoint"). In Figure 3a, the reward value of each waypoint is shown in parentheses after the waypoint number. Given the set of waypoints, the route planner calculates routes for all possible pairs of waypoints going in either direction. A route between any two waypoints is approximated as a set of linear segments between secondary waypoints. This set is translated into a list of tuples $\{\textit{heading}, \textit{distance}, \textit{elevation change}\}$, with each tuple providing instructions on getting from one secondary waypoint to the next.

The route planner uses the Google Maps API (*JavaScript Google Maps Application Programming Interface, version 3.0*, 2014) to calculate the routes. The route planner then considers all pairs of waypoints (in both directions). For each pair of primary waypoints, the Google Maps API is used to identify the secondary waypoints between the primary waypoints. The API provides latitude, longitude, and altitude for the secondary waypoints as a sequence of steps to get from the source waypoint to the destination waypoint. The planner then steps through the secondary waypoints in order and uses the API to determine the heading between the last waypoint and current waypoint. It also calculates the altitude change based on the already retrieved altitudes for the waypoints. The result is a three-dimensional array where the first and second dimension indicate pairs of primary waypoints. The third dimension is used to list routes to secondary waypoints in order, resulting in the aforementioned list of tuples.

3.4. Decision Maker

The decision-making algorithm used in this work, shown in Algorithm 1, is similar to the one presented in (Balaban & Alonso, 2013). It is based on a particle-filtering pattern (Gordon, Salmond, & Smith, 1993) and is summarized below.

Algorithm 1 uses a set of k particles, where each particle p_i is initialized with the starting waypoint wp_1 and assigned a uniform weight of $w_i = 1/k$. The starting waypoint is the waypoint where the vehicle is located at the point of executing the algorithm (not necessarily the original starting point of the route). For simplicity of explanation, the algorithm presented here operates over one-way paths, where the starting waypoint is not always the same as the ending waypoint. In the actual implementation, a choice between one-way and round-trip routes is implemented via a straightforward extension.

During each of the iterations of the algorithm (and for each particle), the path associated with a particle is sampled ran-

[1]This work is restricted to single faults only.

Algorithm 1 PF

inputs: $\{wp_i\}_{i=1}^{N}, K$
outputs: p^*
$p_1, \ldots, p_K \leftarrow \{wp_1\}$
$w_1, \ldots, w_K \leftarrow 1/k$
for $d \leftarrow 1, D$ **do**
  for $k \leftarrow 1, K$ **do**
    $\tau \leftarrow$ *permute*$(\{wp_i\}_{i=1}^{N} - p_k)$
    $l \leftarrow -1$
    repeat
      $l \leftarrow l + 1$
      $p_{test} = \{p_k, \{wp_1, \ldots, wp_l\}\}$
      $\{b, \mathcal{R}, \mathcal{C}_h\} \leftarrow$ *simulate*(p_{test})
      $w_k \leftarrow \{\Theta_R, \Theta_h\} \cdot \{R, -C_h\}^T$
    until $\mathcal{F}(b) = true$
    if $l \geq 1$ **then**
      $p_k \leftarrow \{p_k, \{wp_1\}_\tau\}$
    end if
  end for
  $j \leftarrow \arg\max_j w_j$
  $p^* \leftarrow p_j$
  $\{w_1, ..., w_K\} \leftarrow \{w_1, ..., w_K\} / \sum_{i=1}^{K} w_i$
  $\{p_1, ..., p_K\} \leftarrow resample(\{p_1, ..., p_K\}, \{w_1, ..., w_K\})$
end for

domly out of the set of unvisited waypoints up to the maximum length of N. Each sample is tested in the simulator and the particle weight updated proportionally to the objective function value (which incorporates path costs in addition to rewards). Unless system failure is believed to be likely for even the shortest path extensions, the particle path is extended by one waypoint (the first one in the randomized remaining waypoints set τ).

The number of algorithm iterations, D, is equal to N for the deterministic simulator mode and can be set to $D > N$ otherwise, to help prevent potentially promising particles from being ruled out too early. The highest weight particle is identified and stored after each iteration, to enable interruptibility. Particle weights are then normalized and the particles are resampled. The overall computational complexity of the algorithm is $O(N^2)$.

The objective function used to guide search of the solution space is the following:

$$J = \{\Theta_R, \Theta_h\} \cdot \{R, -C_h\}^T, \quad (1)$$

where $\mathcal{R}$ is the expected cumulative reward along a route, C_h is the correspondent expected health cost, Θ_R and Θ_h are the weights for rewards and health costs, respectively. The simulator used with the PF algorithm utilizes a simplified power consumption model of the rover. A candidate route is divided into linear and turning segments and the resulting list of segments is processed sequentially. For the straight route segments, the following relationship was used to estimate the current drawn from the batteries:

$$i_l = \frac{mgv}{\eta_e E}(\sin\alpha + \mu cos\alpha), \quad (2)$$

where i_l is the linear segment current, η_e is the electrical transmission efficiency coefficient, E is the bus voltage, m is the mass of the rover, g is the acceleration of gravity, v is the magnitude of the linear velocity, α is the incline angle, and μ is the coefficient of surface friction. For this set of experiments linear velocity was kept constant. For the turning segments, a constant rate of turn ω was assumed, associated with a constant current draw i_t. When evaluating a candidate route, a discrete time simulation is performed (with the time step dt normally set to $1s$), taking into account the nonlinear relationship between current draw at a particular instance in time and the corresponding drop in battery cell voltage (and in the SOC of the battery cells).

Table 1. DM model parameters used in the experiments

Parameter	Value	Units
m	150.0	kg
v	0.4	m/s
ω	0.07	rad/s
μ	0.06	
i_t	5.0	A
η_e	0.8	

The battery model used in the simulator is described in (Daigle, Saxena, & Goebel, 2012). The parameters used with the model remained the same as in the aforementioned paper. This equivalent circuit model was integrated and tested with the decision maker prior to the newer electrochemistry model (Daigle & Kulkarni, 2013) becoming available and will be updated to the latter in the near future. The rest of the model parameters used in the experiments are the following are described in Table 1.

A set of $K = 50$ particles was used by PF algorithm. The values of objective function weights used were $\Theta_R = 0.9$ and $\Theta_h = 0.1$.

4. EXPERIMENTS AND RESULTS

The PDM system described above was demonstrated in the field though a set of scenario-based experiments. The details of the scenarios, including the number of waypoints, the overall distance, the distances between waypoints, the fault injection location and fault magnitude were chosen to clearly show the capabilities of the PDM algorithms. The overall distance of the nominal trajectory would have to be such that the vehicle would be capable of completing it fully without system faults. That full nominal scenario trajectory was divided

and arranged into waypoints, chosen to ensure the potential for the system to replan its trajectory before a potential system fault would cause the vehicle to reach the end of useful life. Several fault scenarios were chosen as well, where the PDM system was expected to optimize the trajectory while ensuring the safe return of the vehicle. The location at which the fault is injected and its magnitude were chosen to allow the cumulative effects of the fault to cause the end of the system's remaining useful life before it reached the final waypoint. These experimental scenarios and the results of each are described below.

4.1. Nominal

The nominal scenario consists of 5 waypoints, and no fault was injected during this scenario. The vehicle began at waypoint 9 and traveled to waypoints 2, 5, 7, and back to 9. These waypoints are shown in Figure 3b; the unused waypoints are not shown for clarity. The reward value used for waypoint 9 is 70, and for waypoints 2, 5, and 7 are 30, 90, and 20, respectively. The route planner inserted secondary waypoints as required to navigate this route, where secondary waypoints have no reward.

In this scenario, the vehicle successfully followed the nominal route, covering a distance of approximately 970 m, and gained the reward for all of the waypoints, for a total reward of 280. The nominal route is shown as the blue line in Figure 3b, generated from the vehicle's global positioning system (GPS) sensor values recorded during the scenario. At the end of the nominal scenario the batteries had an estimated SOC of 57.6%. Note that even in the nominal case, the PDM system ran and determined that it was feasible to achieve all of the given waypoints.

4.2. Battery Parasitic Load Fault without PDM

As a second scenario, a battery parasitic load fault (as described in Section 2.1) was injected during the route traversal, with PDM system was not running. This scenario also began and ended at waypoint 9 and consisted of the same waypoints as for the nominal scenario, with the same reward values. However, shortly before reaching waypoint 2, a battery parasitic load fault was injected into the electrical system of the vehicle. This is shown in Figure 3c. The first two relays were activated, resulting in an equivalent parasitic resistance of 21.6 Ω (see circuit diagram in Figure 1) and an increased current draw from the batteries.

In this scenario, with the battery parasitic load active and following the nominal route, the vehicle ran out of power before returning to the starting waypoint. The route is shown in red in Figure 3c, and the location where the vehicle ran out of power is marked. This scenario showed that the nominal route is, in fact, infeasible under the battery drain fault.

4.3. Battery Parasitic Load Fault with PDM

As a third scenario, a battery parasitic load fault was again injected (resulting in the same parasitic resistance of 21.6 Ω) shortly before reaching waypoint 2, while following the same waypoints as the nominal scenario. However, in this case the PDM system was enabled.

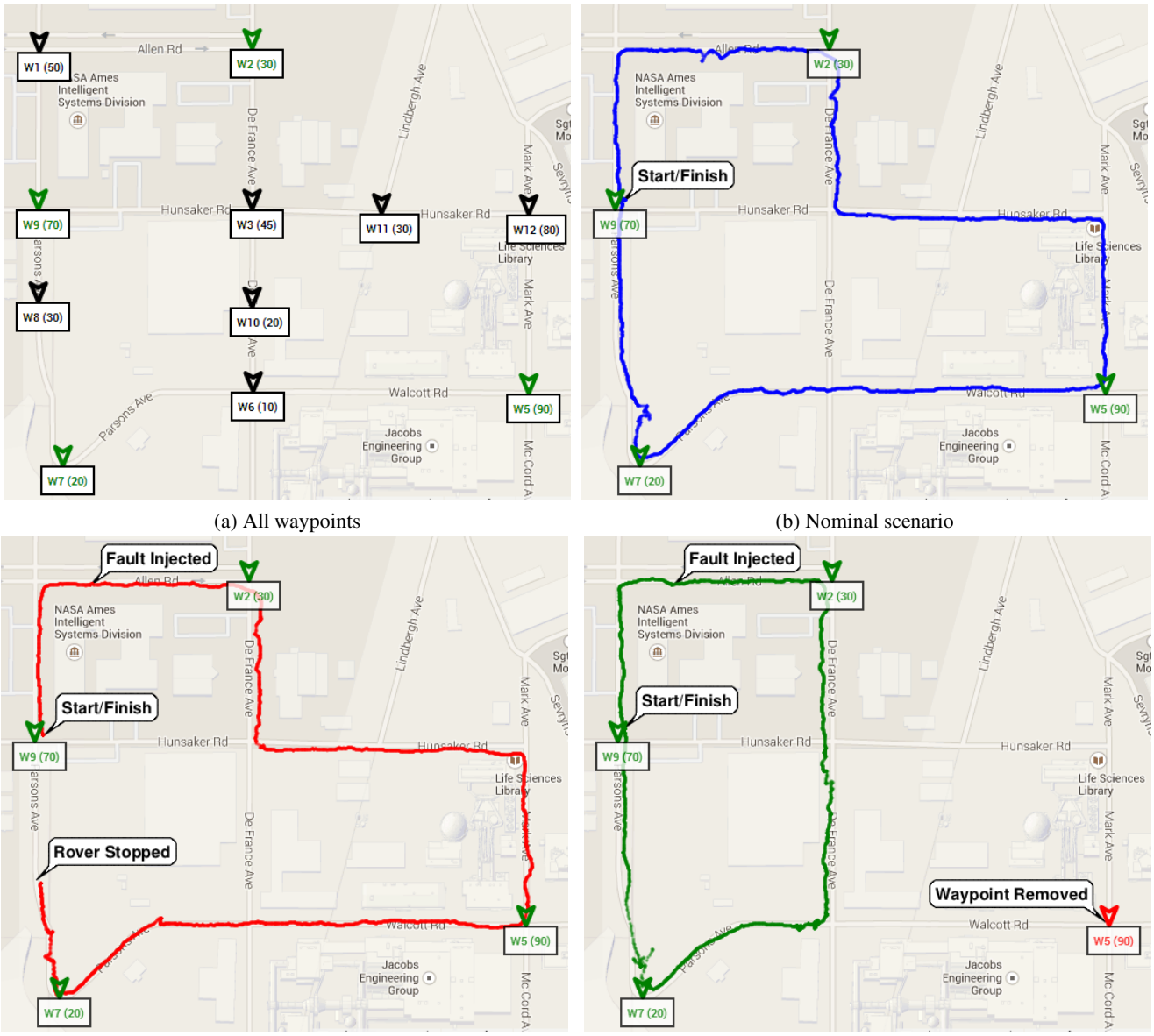
In this scenario, the EPS diagnoser detected that the battery parasitic load has been injected and estimated the equivalent resistance value. It reported its estimate of the equivalent resistance value 14 s after the fault was injected. The estimated resistance was 19.5 Ω, which is an error of only 9.7% from the actual parasitic resistance of 21.6 Ω. The EPS diagnoser then sent that estimated parasitic resistance value to the decision maker along with the battery SOC estimate. When the vehicle arrived at waypoint 2, the decision maker used the information from the EPS diagnoser and determined that the vehicle's original route is no longer feasible. It then performed an optimization to determine a new route which maximized the overall reward for the scenario, while ensuring that the vehicle can return safely to the starting point. As can be seen in Figure 3d, the PDM system eliminated waypoint 5 (shown in red), but kept waypoint 7. The alternative route taken, shown on the figure in green, covered a distance of approximately 713 m. The vehicle successfully navigated the new route and returned to the starting waypoint 9, for a total reward of 190. At the end of the scenario the estimated SOC of the batteries was 14.5%.

Note that a conservative option existed: to return to the starting waypoint as soon as possible after the fault was detected. However, that route would only have gained a total reward of 170. It would not have made optimal use of the vehicle's remaining useful life and, therefore, was not chosen by the PDM system.

4.4. Bus Current Sensor Fault with PDM

As a fourth scenario, a bus current sensor fault was injected (also just before reaching waypoint 2) while following the same waypoints as the nominal scenario. The bus current sensor value was overridden to always report a value of 0.0 A.

In this scenario, the EPS diagnoser detected that the current sensor is faulty and reported that to the decision maker. The decision maker performed an optimization with this vehicle state and the given waypoints and determined that the vehicle is able to complete the original route given the fault. Therefore, it did not modify the vehicle route. Since the mission was unmodified, the vehicle traversed the same route as in the nominal scenario shown in Figure 3b, for the same total reward of 280. At the end of the scenario the estimated SOC of the batteries was 70.8%.



(a) All waypoints

(b) Nominal scenario

(c) Battery parasitic load fault scenario without PDM

(d) Battery parasitic load fault scenario with PDM

Figure 3. Vehicle route taken in nominal and battery parasitic load scenarios with and without PDM

5. CONCLUSIONS

This paper described a successful demonstration of PDM algorithms running onboard a hardware mobile robot test platform. The demonstration required modifications to both the platform and the algorithms. The demonstrations took the form of a set of challenge scenarios. The data files from these scenarios will be made available for download, to allow testing of other prognostic and PDM algorithms.

Planned future work involves deployment of PDM algorithms on an unmanned aerial vehicle, incorporation of uncertainty estimates in the reported health parameters, and implementation of additional faults in the hardware test platform. Possible future work also includes modifications to support different types of decision-making, such as adapting parameters and constraint relaxation in the PDM optimization. As the PDM algorithms are further developed, this robotic test platform will be modified to continue to evaluate them in a real-world setting.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the contributions of researchers and student interns at NASA Ames Research Center: Kai Goebel, Brian Bole, Shankar Sankararaman, and Sebastian Hening. The funding for this research is

provided by NASA ARMD System-wide Safety & Assurance Technology (SSAT) project.

REFERENCES

Balaban, E., & Alonso, J. J. (2013). A Modeling Framework for Prognostic Decision Making and its Application to UAV Mission Planning. In *Annual conference of the prognostics and health management society* (pp. 1–12). New Orleans, LA.

Balaban, E., Narasimhan, S., Daigle, M., Celaya, J., Roychoudhury, I., Saha, B., ... Goebel, K. (2011, September). A mobile robot testbed for prognostics-enabled autonomous decision making. In *Annual conference of the prognostics and health management society 2011.* Montreal, Canada.

Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., & Gorospe, G. (2013). Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *International Journal of Prognostics and Health Management*, *4*(1).

Balaban, E., Saxena, A., Bansal, P., Goebel, K. F., & Curran, S. (2009). Modeling, detection, and disambiguation of sensor faults for aerospace applications. *Sensors Journal, IEEE*, *9*(12), 1907–1917.

Balaban, E., Saxena, A., Narasimhan, S., Roychoudhury, I., Goebel, K., & Koopmans, M. (2010, September). Airborne electro-mechanical actuator test stand for development of prognostic health management systems. In *Annual conference of the prognostics and health management society 2010.* Portland, OR.

Daigle, M., Bregon, A., Biswas, G., Koutsoukos, X., & Pulido, B. (2012, August). Improving multiple fault diagnosability using possible conflicts. In *Proceedings of the 8th ifac symposium on fault detection, supervision and safety of technical processes* (p. 144-149).

Daigle, M., Bregon, A., & Roychoudhury, I. (2014). Distributed prognostics based on structural model decomposition. *IEEE Transactions on Reliability*.

Daigle, M., Koutsoukos, X., & Biswas, G. (2009, July). A qualitative event-based approach to continuous systems diagnosis. *IEEE Transactions on Control Systems Technology*, *17*(4), 780–793.

Daigle, M., & Kulkarni, C. (2013, October). Electrochemistry-based battery modeling for prognostics. In *Annual conference of the prognostics and health management society 2013* (p. 249-261).

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., & Poll, S. (2010, September). A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *IEEE Transactions of Systems, Man, and Cybernetics, Part A*, *4*(5), 917–931.

Daigle, M., Roychoudhury, I., & Bregon, A. (2013, October). Qualitative event-based diagnosis with possible conflicts: Case study on the fourth international diagnostic competition. In *Proceeedings of the 24th international workshop on principles of diagnosis* (p. 230-235).

Daigle, M., Saxena, A., & Goebel, K. (2012). An Efficient Deterministic Approach to Model-based Prediction Uncertainty Estimation. In *Annual conference of the prognostics and health management society.*

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, *140*(2), 107–113.

Henning, M. (2004). A new approach to object-oriented middleware. *IEEE Internet Computing*, *8*(1), 66–75.

JavaScript Google Maps Application Programming Interface, version 3.0. (2014). Mountain View, California: Google Corporation.

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

LabVIEW version 12.0.0.4029. (2012). Austin, Texas: National Instruments Corporation.

Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *29*(6), 554-565.

Poll, S., de Kleer, J., Abreau, R., Daigle, M., Feldman, A., Garcia, D., ... Sweet, A. (2011, October). Third international diagnostics competition – DXC'11. In *Proc. of the 22nd international workshop on principles of diagnosis* (pp. 267–278).

Poll, S., Patterson-Hine, A., Camisa, J., Nishikawa, D., Spirkovska, L., Garcia, D., ... others (2007, May). Evaluation, selection, and application of model-based diagnosis tools and approaches. In *Aiaa infotech@aerospace 2007 conference and exhibit.*

Reveley, M. S., Kurtoglu, T., Leone, K. M., Briggs, J. L., & Withrow, C. A. (2010, December). *Assessment of the state of the art of integrated vehicle health management technologies as applicable to damage conditions* (TM No. 2010-216911). Cleveland, OH: NASA.

Roychoudhury, I., Daigle, M., Bregon, A., & Pulido, B. (2013, March). A structural model decomposition framework for systems health management. In *Proceedings of the 2013 ieee aerospace conference.*

Smith, M., Byington, C., Watson, M., Bharadwaj, S., Swerdon, G., Goebel, K., & Balaban, E. (2009). Experimental and analytical development of health management for electro-mechanical actuators. In *Ieee aerospace conference* (pp. 1–14).

BIOGRAPHIES

Adam Sweet is a research engineer in the Diagnostics and Prognostics group at NASA Ames Research Center. He graduated with an MS in Mechanical Engineering from UC Berkeley in 1999, and has worked at Ames ever since. His project experience encompasses robotics, hybrid system simulation, model-based diagnosis, and flight software development for nanosatellites.

George Gorospe rreceived the B.E. degree in Mechanical Engineering from the University of New Mexico, Albuquerque, New Mexico, USA, in 2012. Since October 2012, he has been a research engineer at NASA Ames Research Center. In May 2013 he joined Stinger Ghaffarian Technologies and the Prognostic Center of Excellence at NASA Ames Research Center. His current research interests include space mission design, systems engineering, and autonomous mobile robot control and control systems design.

Matthew J. Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

José R. Celaya is a research scientist with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Electrical Engineering in 2003, all from Rensselaer Polytechnic Institute, Troy New York; and a B. S. in Cybernetics Engineering in 2001 from CETYS University, México.

Edward Balaban is a research engineer in the Diagnostics and Prognostics group at NASA Ames Research Center. He received the Bachelor degree in Computer Science from The George Washington University in 1996 and the Master degree in Electrical Engineering from Cornell University in 1997. His main areas of interest are diagnostics, prognostics, and prognostics-enabled decision making. During his years at Ames he has participated in the research and development of system health management elements for the X-34 experimental reusable launch vehicle, the International Space Station, robotic astronaut assistants, autonomous planetary drills, and other aerospace applications. He is a member of the PHM Society, the IEEE, and the AIAA.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.

Sriram Narasimhan is a Project Scientist with University of California, Santa Cruz working as a contractor at NASA Ames Research Center in the Discovery and Systems Health area. He received his M.S and Ph.D. in Electrical Engineering and Computer Science from Vanderbilt University. He also has a M.S in Economics from Birla Institute of Technology and Science. His research interests are in model-based diagnosis with a focus on hybrid and stochastic systems. He is the technical lead for the Hybrid Diagnosis Engine (HyDE)

Gear Diagnostics – Fault Type Characteristics

Alexander Bliznyuk[1], Ido Dadon[2], Renata Klein[3] and Jacob Bortman[4]

[1,2,4] *Laboratory for Health, Department of Mechanical Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
bliznyuk@post.bgu.ac.il

[3] *R.K. Diagnostics, Gilon, Israel*
Renata.Klein@RKDiagnostics.co.il

ABSTRACT

To date, the majority of existing Condition Indicators for gears are based on various statistical moments of a recorded time history. A supplementary analysis proposed in this study, shall suggest an approach that may, in the future, enable the identification of faulty gearwheel and possibly fault type in the system. In this work, a combined analytical and empiric approach is applied. This approach is based on the assumption that reliable dynamic models can be utilized to predict the effects of faults on vibrational patterns. Dynamic model generated signatures are used to verify experimental findings. Moreover, discrepancies between simulated and actual results, combined with understanding of the assumptions and omissions of the model, are helpful in understanding and explaining the experimental results.

A spur gear transmission setup was used for experiments, along with an electric AC motor and a friction belt loading device. The experimental runs were conducted at varying speed settings. Two types of faults, a tooth face fault and a tooth root fault, were seeded in the experimental transmission and into the model. The effect on extracted signal features is examined.

The purpose of this study is to evaluate fault detection capabilities of proposed diagnostic tools at the presence of two seeded faults of varying severity, verified by a dynamic model. Observed differences between examined fault types and their manifestation will be discussed. A basis for future work on prognostics capabilities is laid by a varying degree of tooth root fault.

1. INTRODUCTION

Most existing Condition Indicators (CI) for gears are defined by a statistical analysis of various signals in time or cycle domains (Dempsey, Lewicky and Le, 2007; Lewicky, Dempsey and Heath, 2010). Most of these CI are various modifications of statistical moments (RMS, Kurtosis etc.). When applied to a gear pair time or cycle history, statistical CI differentiate between signals originating in undamaged and damaged gear pairs, but a difficulty in distinguishing between types of faults and fault location exists. In this work, an analysis of side bands of gear meshing frequencies is suggested as a tool for evaluation of gear health. Side bands analysis was proposed in other works as a tool for fault identification, and classification of side band groups was defined by Klein (2012).

This work aims to show that a more detailed analysis of faults in gears can be harvested in the order domain. In this work a concept of a division of a fault effect into two aspects, 'dynamic' and 'structural', is introduced as a possible explanation of several observed differences between faults.

Simulated vibration signals from a dynamic model, developed in the BGU HUMS lab, are compared with experimental results to help further understand the latter. Currently, the model is qualitative and purely dynamic, which means it does not account for the transmission path of the signal from its origin to the sensor.

2. EXPERIMENTAL SETUP

2.1. Setup

A simple one stage spur gear system was used in this research (figure 1). The main advantage of such a setup over a real life complex transmission lies in easier interpretation of results, for better understanding of the basic physics of this problem.

Standard (evolvent) profile spur gear pair, of module 2.5 [mm] was used, 17T driving gear (pinion) and 49T driven gear. The pinion is seated on the "In" shaft. The transmission reduces the speed of the "Out" shaft containing the driven gear and the loading device. Both shafts are supported by two ball bearings each.

Alexander Bliznyuk et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The experimental setup is driven by a 3 phase asynchronous AC induction motor. An open loop controller is used to set the frequency input of the motor. An optical encoder (24 band/revolution) is used to record "In" shaft RPS during the run.

The setup is torque-loaded via a friction belt-wheel pair. The belt is tensioned by a selectable amount of weights. The resulting side effect of bending of the shorter "Out" shaft due to radial stress is negligible.

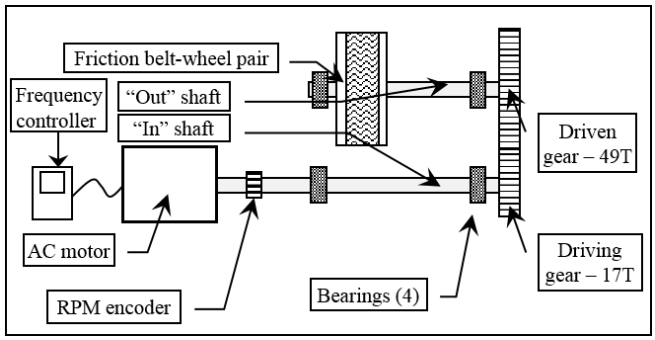


Figure 1. Experimental setup (schematic).

A Dytran tri-axial accelerometer was used to measure the vibration in the proximity of the gear mesh point. The accelerometer was fixed below the pinion, with the X axis aligned as the tangent direction at the gear mesh point, Y as the radial direction and Z as the axial direction (see figure 2).

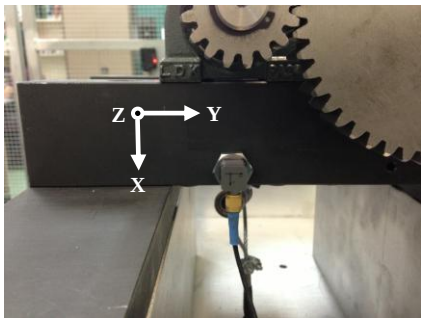


Figure 2. Accelerometer location and orientation.

2.2. Experiment Conduct

Experiments were conducted for each of six configurations: undamaged transmission ("Healthy"), a gear carrying a tooth face fault ("spall"), cracked pinion ("P*I*"), and three degrees of cracked gear ("G*I*", "G*II*", "G*III*"). For each configuration, 20 experimental runs were performed, at four varying loadings of the friction belt and at five AC motor input frequency settings.

2.3. Seeded Faults and Seeding Methods

Two faults were selected for seeding in the study. The faults selected simulate common and essentially different real life faults, relatively simple to simulate both in the experimental and model environments. A tooth face defect was seeded in the gear, simulating a fault of the spall/pitting type. The single tooth defect (figure 3) was seeded by a removal of material from the tooth face at a portion of the tooth's width. In similarity to the effect of a common spallation (or pitting) on tooth meshing, the presence of the fault reduces the contact stiffness of the tooth, but does not yet alter the general evolvent profile of the tooth.

A crack was seeded in the root of a single tooth, simulating a fatigue crack. The fault was seeded by EDM (Electrode Discharge Machining) at three fault severity degrees (crack depth of 1.4, 2.1 and 3.5 mm of total tooth width of 4.8 mm) in the gear (figure 4) and at the first degree only in the pinion.

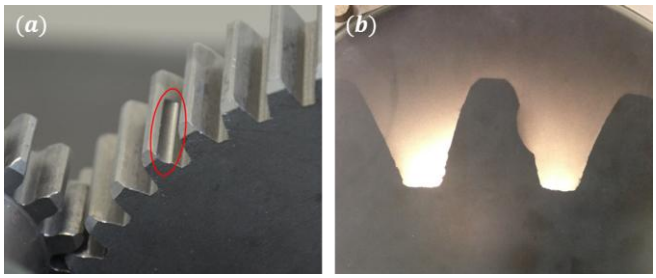


Figure 3. Seeded spalling defect (a) encircled; (b) view of the damaged tooth.

In this work, a tooth flaw is considered to have a dual effect on vibration signature. The "dynamic" component of the flaw affects the gear meshing at the point of defect, altering the dynamics behind the generated acceleration signal. The "structural" flaw alters the transmission path from the acceleration origin (gear mesh point) to the sensor.

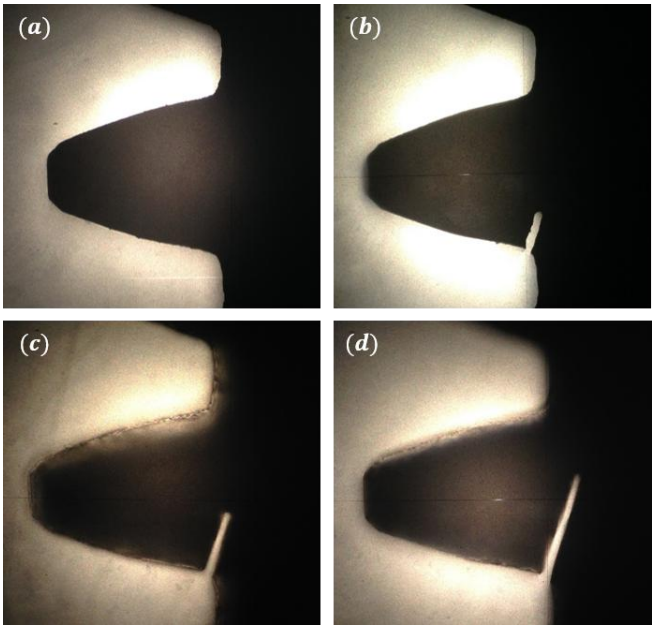


Figure 4. Seeded tooth root crack (a) healthy; (b) 1.4 mm; (c) 2.1 mm; (d) 3.5 mm.

3. DATA ANALYSIS

Experimentally obtained signals were analyzed in the workflow depicted in figure 5. The raw signal was resampled into the cycle domain, and then synchronously averaged. The resulting signal was then mapped into the order domain, and features were extracted from the PSD (Power Spectrum Density).

3.1. Angular Resampling

As for all realistic revolving machinery, speed (RPS) was only approximately constant during the experimental runs, with relatively slight deviations from a mean value. The resulting signal is classified as non-stationary, and has "smeared" spectral contents due to the non-constant frequency of the signal periodic components. To allow for an accurate representation in the order domain, angular resampling was applied to the signal's time history.

During angular resampling, the signal is resampled by constant rotation angle (cycle) increments rather than constant time increments as recorded originally. Signals that undergo angular resamping are said to be transferred from the 'time' domain to the 'cycle' domain. Simulated signatures (results of the dynamic model) are by definition of absolutely constant input RPS (classified as deterministic periodic signal), and therefore do not undergo this part of the processing.

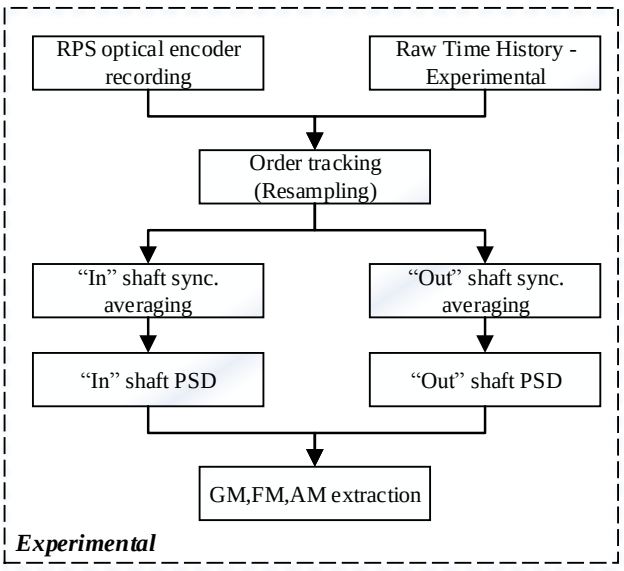


Figure 5. Experimental data analysis workflow.

3.2. Time Synchronous Averaging (TSA)

The recorded data contains a substantial amount of data unrelated to the process of gear meshing. The purpose of synchronous averaging is the removal of all signal components asynchronous with the phenomena examined, such as bearing tones, noise etc. Two TSA were calculated for each signal, by the "In" and by the "Out" shaft speeds.

3.3. Calculation Error

RPS (Revolution per Second) measurement and decoding accuracy is a major error factor of calculated synchronous average and consequent PSD features. Inaccurate RPS causes smearing of PSD peaks due to averaging out of synchronous data, through inaccurate angular resampling. Although helpful with removal of noise and asynchronous components, a large amount of averaged cycles increases this error. To minimize differences between signatures, the length of measured data was set to be a constant amount of machine cycles (200) rather than a constant time interval.

3.4. Order Feature Extraction

The synchronously averaged signals were mapped from the cycle domain to the order domain by a windowed Welch's periodogram. From the PSD (Power Spectral Density), three features were calculated.

The gear mesh order is the z^{th} shaft harmonic (where z is number of teeth on shaft's gearwheel). The sum of the first five harmonics of gear mesh amplitude in the PSD was defined as the GM feature. The GM is assumed to carry the energy resulting from the meshing of all (defective and healthy) teeth. The GM is identical whether it is calculated from the "In" or the "Out" shaft synchronous average.

$$GM \equiv \sum_{h=1}^{5} \hat{x}\left(h \cdot z \cdot f_s\right) \cdot df \tag{1}$$

As described in other publications (Klein, 2012), sidebands (SB) in the order domain on both sides of the main gear mesh frequency are caused by the amplitude and frequency modulations of the shaft speeds. These take the form of accompanying pairs of peaks, at constant spaces (equal to the modulating wave frequency), as can be seen in the example in figure 6. Two types of sidebands were observed in all signatures – those associated with the "In" shaft and those associated with the "Out" shaft.

Sidebands groups that were considered in this study as features are:

- AM (Amplitude Modulation) – the sum of amplitudes of the first two (as defined by Klein in 2012) pairs (n=1 to 2) of SB around a GM harmonic:

$$AM \equiv \sum_{h=1}^{5} \sum_{n=1}^{2} \hat{x}\left(\left(h \cdot z \pm n\right) \cdot f_s\right) \cdot df \tag{2}$$

- FM (Frequency Modulation) – sum of all the other available SB amplitudes that can be associated with the GM harmonic. The association limit in the order domain was set to be mid-way between adjacent GM harmonics (*n=3 to z/2*):

$$FM \equiv \sum_{h=1}^{5} \sum_{n=3}^{z/2} \hat{x}\left(\left(h \cdot z \pm n\right) \cdot f_s\right) \cdot df \qquad (3)$$

Each feature (FM, AM) was calculated by the summation of all related peak amplitudes for the first five harmonics (denoted h) of GM.

Individual peaks in the spectrum are associated with dynamic effects that originate from the machine rotation. Therefore they occur at discrete frequencies, which are multiplications of the machine rotation speed. Transmission path is composed of structural effects that are not dependent on rotation. Transmission attenuates or amplifies the dynamic peaks and all other frequencies, and is continuous. The curve in the spectrum (Klein, 2013) which represents the transmission path is illustrated in figure 7.

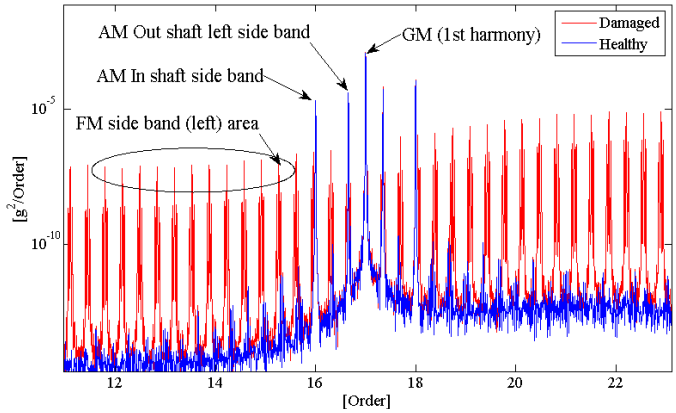


Figure 6. Example of GM, FM, and AM manifestation in PSD of a pair of simulated runs (with\without flaw). In example shown, seeded fault can be observed in "Out" shaft FM sideband increase.

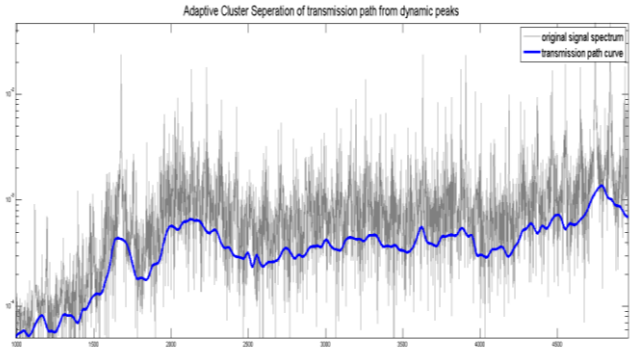


Figure 7. Example of the manifestation of dynamic reciprocating effects as peaks (grey) over a general transmission path spectral curve (blue)

3.5. Cycle Domain Analysis

RMS and kurtosis were calculated for both synchronously averaged cycle domain (resampled) signals (by In and Out shafts). These were calculated both for the complete signal and for the residual (as defined by Dempsey et al, 2007). These moments are currently the basis for most common Condition Indicators for gears.

3.6. Spherical Coordinates

It is assumed that the transmission function alters both the magnitude and direction of the generated vibration. A spherical coordinates approach is proposed in this study (equation 4). Among the advantages of this approach is the measurement of fault effect on vibration magnitude, rather than one dimensional vibration changes which are an incomplete representation of the fault manifestation.

Spherical magnitudes were calculated from the tri-axial signal (equation 4).

$$\left\|\bar{a}\right\| \equiv \sqrt{a_x^2 + a_y^2 + a_z^2}, \quad \bar{a} \in R^3 \qquad (4)$$

The same data analysis that was performed for the recorded separate axis was repeated for the vector magnitude of the spherical coordinates.

Spherical magnitudes analysis allows the consideration of vibration magnitude only, detached from vibration direction.

4. DYNAMIC MODEL

Following the procedure described in our previous article (Dadon et al 2014), a qualitative dynamic model of a spur gear transmission is developed in order to describe the dynamic vibration response of the experimental gearbox system. The following description of the model is concise, since the modeling is not the primary subject of this article.

The experiment system (figure 1) was idealized and all of its components were incorporated in the dynamic model, as shown in the scheme in figure 8. A constant input velocity and external applied load are the boundary conditions, which are chosen to simulate the experimental settings.

The interaction of a gear pair is modelled by linear springs with a varying mesh stiffness, which is dependent on the angular position of the gears. The stiffness of a spur gear tooth is determined by considering the strain energy, Hertzian contact and gear body-induced tooth deflection due to contact of teeth (Chaari, Baccar, Abbes and Haddar, 2008; Chen & Shao, 2011). Two directions of transverse displacements are examined, radial and tangential.

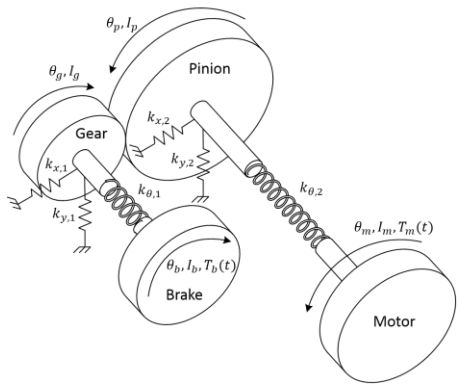


Figure 8. Spur gear system model

The coupled differential equations of the non-linear multi degree of freedom (MDF) system, describing the motion of the specified system, are derived from the Euler-Lagrange equations. The general form of the equations is therefore

$$[M]\{\ddot{u}\}+[C]\{\dot{u}\}+ f_S(\{u\})=\{F_{ex}(t)\} \quad (5)$$

Where u is the vector of generalized coordinates, given by

$$u=\{x_1, y_1, x_2, y_2, \theta_g, \theta_p, \theta_m, \theta_b\}^T \quad (6)$$

$[M]$ is a diagonal mass matrix, $\{F_{ex}(t)\}$ is the external excitation force vector and $f_S(\{u\})$ is a non-linear relative displacement function. The non-linearity is a result of the structural stiffness matrix, particularly the variable gear mesh stiffness. The coupled differential equations are solved using Newmark's numerical method (Chopra, 2001).

The effect of described seeded tooth faults on the dynamic response is expressed by alteration of the gear mesh stiffness. The geometric form (type, location and size) of a fault defines the gear mesh stiffness alterations as function of mesh angle. In this manner only the dynamic effects of a fault are considered. Since the transmission path effects between the signal origin and the sensor are not modelled, structural effects of a fault are not accounted for.

Results obtained via the dynamic model are titled in this paper 'simulated results'.

5. RESULTS

Both experimental and simulated (model) results are described in this chapter. Typical results are displayed in the following figures. It was found that available loading capability in current experimental setup is negligible when compared with the effects of varying rotation speed, therefore all charts are displayed as a function of varying RPS.

5.1. Statistical Moments (Cycle Domain Analysis)

Spall fault was manifested in RMS increase in both residual and ordinary TSA signals of the In shaft. The spall fault was not manifested in kurtosis.

First degree of gear crack was not detected by cycle domain analysis (RMS and kurtosis). The pinion crack was detected primarily by an increase in RMS of the Out shaft, more pronounced in the residual signal.

Second and third degree gear cracks were very similarly detected by an increase in RMS of both shaft.

To conclude, except gear crack I, all faults were detected by RMS. Kurtosis remained unchanged by all types of seeded faults.

5.2. Order Features

The GM feature was not found to be a good fault indicator, but was the prime reactant to load changes, in good accordance with the simulated results. The AM feature was anticipated by the simulated results to be a secondary fault indicator, but in experiments was overwhelmingly affected by shaft imbalance and gear eccentricity. This was also verified by simulated runs with increased imbalance. FM seems to be the primary feature for consideration.

5.3. Fault Detection

Two types of seeded faults were studied, as described in chapter 2. The spall fault was seeded in the gear. A tooth root crack was seeded in both the gear and the pinion (separate experiments). It was observed that all faults were detected primarily in the FM feature.

As can be observed in figure 9, all seeded faults (gear spall, gear crack, pinion crack) cause a significant change in FM (In, Out or both). The most noticeable fault proved to be a pinion tooth root crack, with a significant increase of FM Out (Tangential).

Spall fault caused an increase of In shaft FM. A minor increase of FM Out was observed.

Generally, all fault manifestation increased with growing RPS. The dominant axis for fault manifestation was the tangential, thus chosen for display in all figures.

In simulated results, similar curves of FM increase vs. RPS were calculated. Simulated FM of shaft not carrying the faulty gearwheel (e.g., Out for PI) exhibited the same behavior but at substantially lower amplitudes (thus indiscernible in figure 10).

5.4. Fault Type Diagnosis

As can also be seen in figure 9, a variety in FM response to seeded flaw exists. For example, substantial increase in FM

Out is associated with crack (pinion or gear) and not with spall.

A more significant difference between crack and spall faults is in the overall spectrum curve, representing the structure effects. The spall fault has a minor structural manifestation, and is mostly a dynamic fault, whereas cracks have a significant structural effect on the vibration signal travelling from the origin to the sensor. Therefore crack faults alter the transmission path more than spall faults. This alteration of the transmission path (and as a result, the overall spectrum curve) may offer a tool to differentiate between the two fault types.

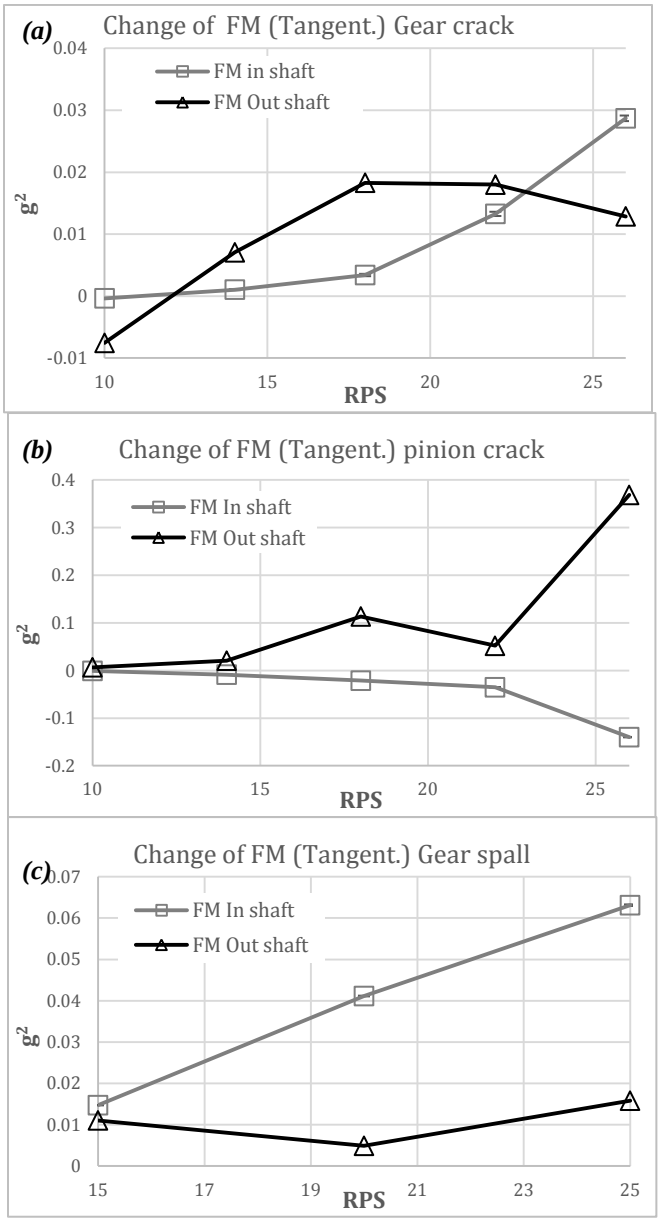


Figure 9. Increase or decrease of Experimental FM Tangential (In & Out) as a result of, (a) gear crack, (b) pinion crack, (c) gear spall

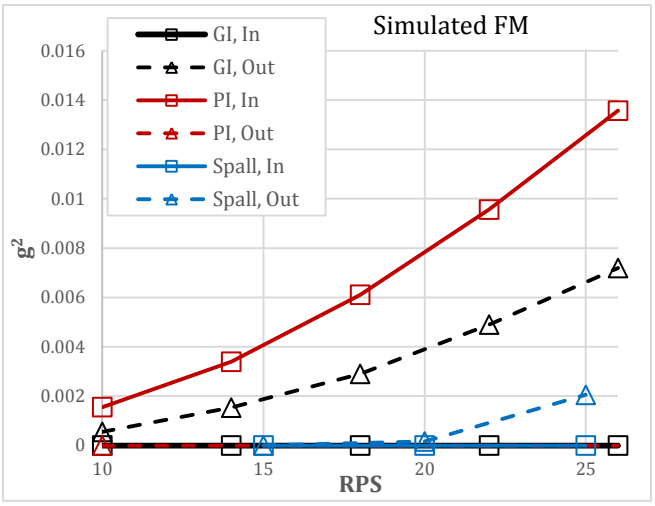


Figure 10. Increase of Simulated FM (In & Out) as a result of Spall, gear crack and pinion crack.

In figure 11(a), a damaged (spall) and healthy frequency domain PSD of similar RPS and load conditions are shown (tangential axis). The underlying transmission function curves of the healthy and damaged signatures are similar, and the main differences are in the side bands amplitudes of the 1st, 2nd and 3rd harmonics of the gearmesh frequencies. In comparison, in figure 11(b) four runs (healthy, GI , GII, GIII) are shown. In this case, the transmission functions vary significantly, with major differences arising above 850 Hz. Since the only difference between the runs is the severity of the crack, the fault effect on transmission function (expressed by overall spectrum curvature) is hereby shown.

5.5. Identification of Faulty Machine Gear Wheel

In this work, identification of faulty machine gear wheel is not achieved. Nevertheless, a suggestion arises as to a possible research direction for identification of fault location.

Crack location (gear/pinion) may be deduced from the effects of the structural aspect of the fault. As already discussed, FM related to shaft carrying the faulty wheel is attenuated in comparison with FM of shaft not carrying the faulty wheel. As can also be seen in figure 9, for higher RPS (22,26) FM related to faulty shaft is expressed in downward sloping (concave) curve, while the healthy shaft's FM has an upward sloping (convex) curve response. While the latter fits curvature predicted by simulations for all faults at all locations, the former does not.

FM feature is extracted at specific (constant) locations in the order domain, while the system transmission function is constant in the frequency domain. Changing RPS causes a shift of the order domain in relation to the frequency domain. Ergo, curves of FM as function of RPS depend on transmission function. A change in these curves due to

introduction of a fault suggests an alteration of transmission function by the seeded (structural) fault.

Signals that travel from the mesh point (vibration origin) through the healthy gearwheel are unaffected by the crack, while signals travelling through the cracked gearwheel experience a modified transmission path due to the crack. This suggests that FM In is not affected by the structural element of the crack, while FM Out is.

The nature of FM curve as a function of RPS is a property of initial (healthy) transmission and machine in question, and is therefore a case specific phenomenon. It may be possible to differ between gear and pinion cracks in this manner in the future, but further study and modeling of the transmission function is required to generalize and verify this special case observation.

5.6. Fault Severity

The fault is seeded in the gear, seated on the Out shaft. As can be seen in figure 12(a), a gradual increase in FM In side bands is obvious as tooth root crack propagates, making it possible to assess fault severity levels. All curves exhibit similar RPS dependency of a rising slope (concave). FM In dependence on RPS fits simulated results for FM Out (figures 12, 13).

- FM Out response to fault severity is a notable increase from healthy to GI, an additional increase to GII, and an unexpected drop in values for the maximal severity GIII.
- FM Out (RPS) curves are of a different (convex) nature, especially for GIII.

Both these properties of the FM Out were not anticipated by the simulated results and are not observed in FM In. As explained in chapter 5.5, this may be reasoned by the structural effect of the fault on the transmission function.

In cycle domain analysis, residual signals are dominated entirely by the FM feature (with GM and AM removed). As to be expected, very similar figures regarding crack severity were achieved by a calculation of the RMS of the residual of synchronously averaged signals (by In and Out shafts).

Simulated FM In response to crack on the Out shaft gear was very similar to simulated FM Out (shown in figure 13), at lower amplitudes.

6. DISCUSSION

6.1. Order Domain Analysis Capabilities

In all acquired results, the FM feature was the most reliable indicator of the presence of a seeded fault. All faults were readily discernible in a change of FM. Varying load had a less significant effect on FM increase, perhaps due to limited loading capability of available apparatus. It was shown that higher RPS produces significantly better fault expression in FM, in accordance with simulated results.

Distinguishing between different faults and fault location may be accomplished in the future by observations in the order domain as depicted in chapter 5. This requires further study of the transmission function alteration by the seeded fault ('structural' aspect of the fault), and additional study cases before any definitive conclusions can be made.

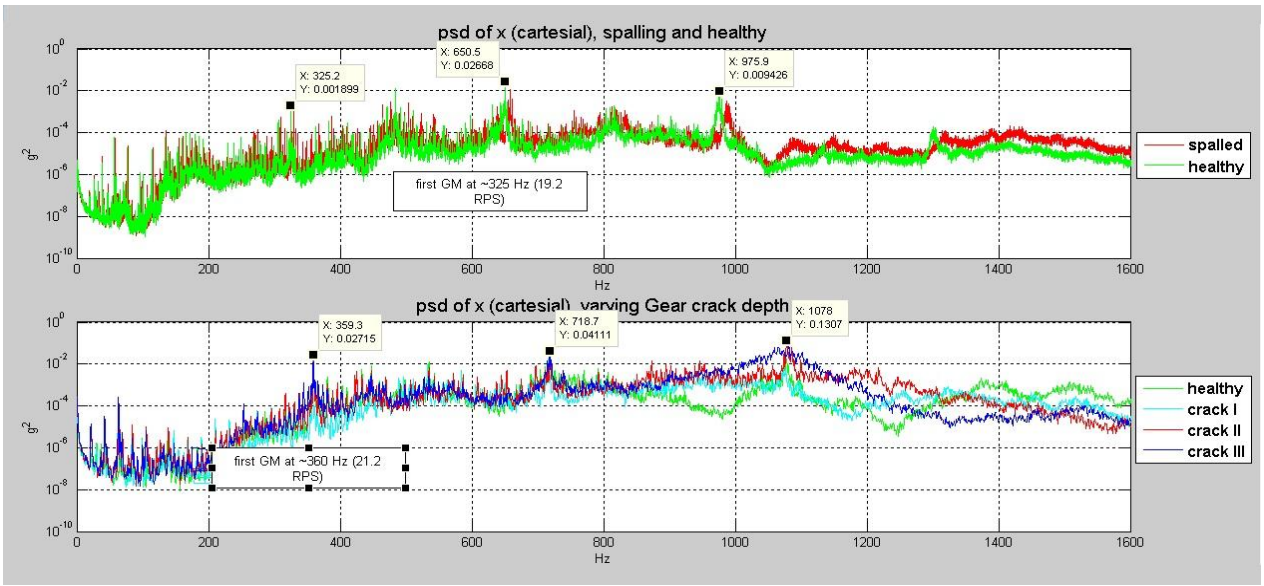


Figure 11. PSD (tangent.) of frequency domain, (a) spall vs. healthy, (b) various degrees of cracked gear

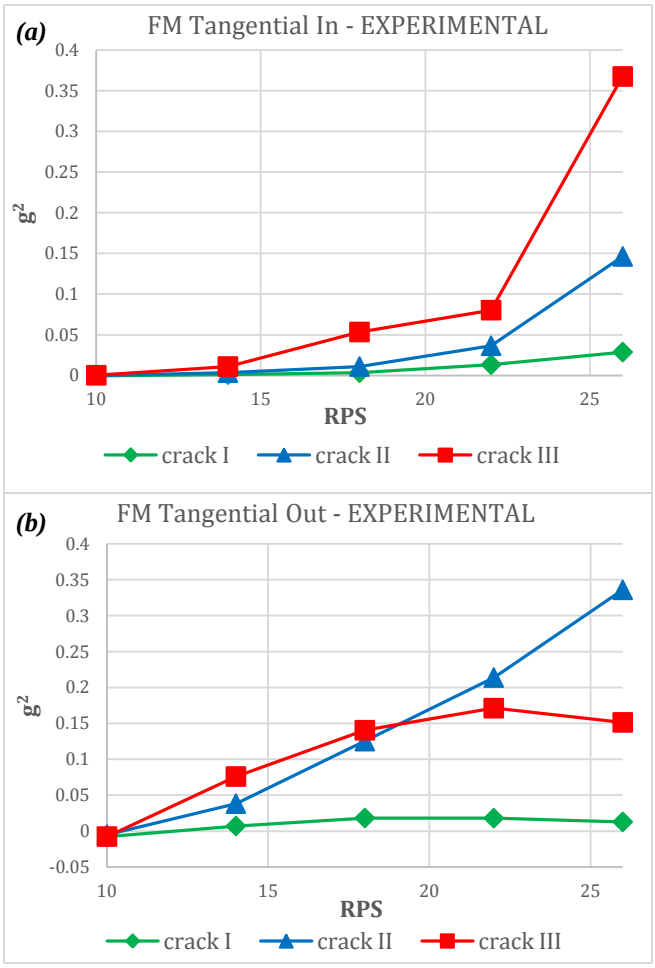


Figure 12. Increase of FM Tangential (a) In (b) Out at three levels of crack (gear).

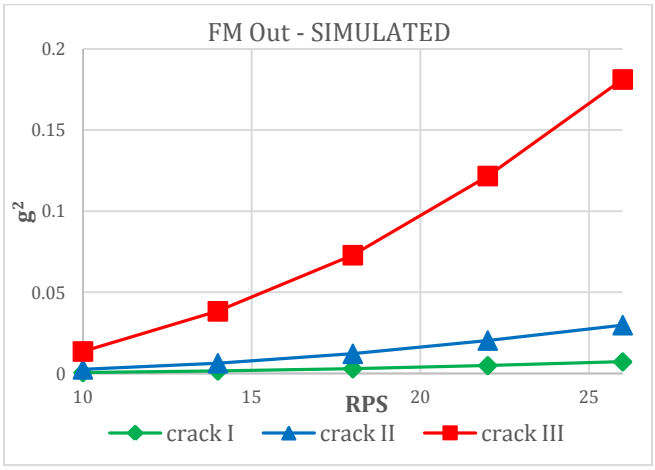


Figure 13. Increase of FM Out at three levels of crack – Simulated results.

A growing severity of a gear tooth root crack was manifested in FM In (not fault carrying shaft), with values of side band increasing in an obvious correlation to crack size. The fault carrier shaft (Out) showed an unexpected drop of FM for crack III (figure 12(b)).

6.2. Simulated results comparison

Two discrepancies are observed in the simulated versus experimental results.

FM of shaft not carrying the fault is almost idle in the simulated results. In the actual measurements FM of both shafts was affected by the fault. Crack deepening causes a similar response in FM Out (figure 13) as observed in the other shaft in experimental results (figure 12(a)). A coupled response of both shafts to all faults is observed in experiments. This coupling is much weaker in the dynamic model equations.

Current version of the dynamic model does not account for the effects of transmission function on the dynamic response of gear meshing. Furthermore, the alteration of transmission function caused by faults is not included in the model. In regards to the distinction between dynamic and structural faults, the model currently deals with the 'dynamic' component only. It is likely that most of the discrepancies between simulated and actual results are explained by this deficiency.

6.3. Spherical vs. Cartesian Coordinates

Most of the extracted features and trends discussed were visible in the Cartesian (tangent, radial, axial) separate axis analysis, but crack fault manifestation was not consistent: some experimental runs showed an increase in tangential, or radial axis, with no obvious pattern as to which axis responds to the fault and under which conditions. In several runs, only one or two out of the three axis responded to the fault.

Representation in spherical coordinates (vibration vector magnitude analysis) enhanced the results and improved consistency and similarity between runs, with overall magnitude FM behaving in a consistent manner over varying RPS (examples in figures 14, 15).

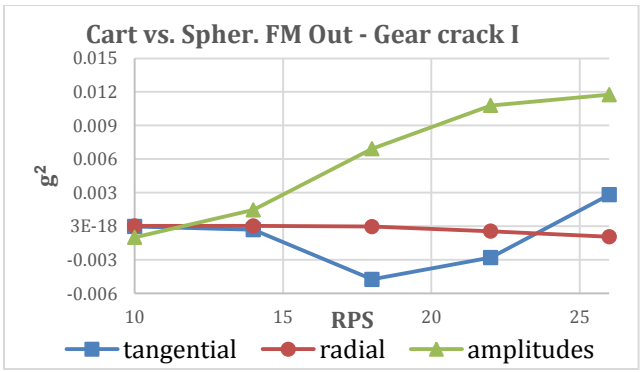


Figure 14. Spherical and Cartesian coordinates FM Out. Showed are FM Out sums related to #1 harmonic only.

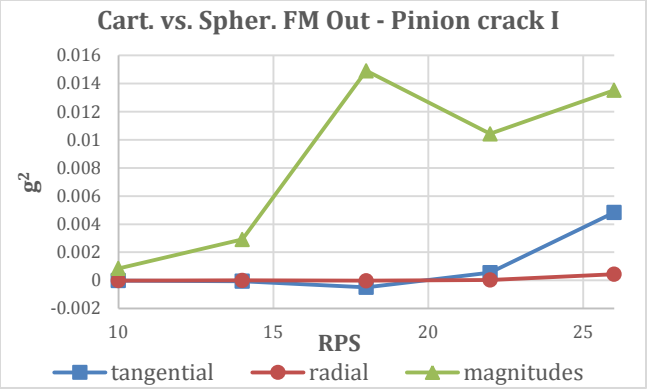


Figure 15. Spherical and Cartesian coordinates FM Out. Showed are FM Out sums related to #1 harmonic only.

In the scope of this work, only magnitudes were considered and analyzed. Some information is lost in the transition from Cartesian coordinates, specifically the effect of fault on vibration vector orientation.

A possible solution to the specified problem, and a subject of further research may be the same spectral analysis applied to an angular property of the acceleration vector.

In figure 16 the same information as in figure 11 is shown for the spherical magnitudes. It can be seen that transmission function of the magnitudes is less affected by the introduction of gear crack than the transmission function in the tangential direction only. This suggests that the alteration to the transmission function in shown bandwidth is mainly in changing the direction of the vibrating signal and not by introduction of natural frequencies (local amplifications of vibration). Attenuation of tangential signal for a certain frequency, for example low amplitudes for healthy tangential signature around 950 Hz, is compensated by high radial and/or axial amplitudes around 950 Hz, and thus spherical magnitude is unaffected. This hints to the possible importance of the analysis of acceleration (unit) vector direction oscillation.

7. CONCLUSION

Order domain features, and specifically FM, may be utilized as a supplementary or even a leading fault indicator. Crack size seems to be directly correlated with FM side bands energy.

The separation of fault effect on vibrations to 'structural' and 'dynamic' components was defined. The same approach may be utilized in the analysis of the signal. An extraction of the transmission path curve from the PSD may allow for a separate analysis of fault effect on transmission ('structural') and on features extracted from a PSD without a transmission function ('dynamic'). The features calculated in this work were not separated in this manner, and the effects of one and the other intertwined.

A deeper understanding and analysis of the 'structural' effects of a flaw may lead to better discrimination between types of faults and identification of faulty gear wheel.

Current simulated results are purely 'dynamic', as explained in chapter 6. A Finite Discrete Element scheme or another numeric supplementary tool can be used to simulate the 'structural' aspect, to achieve a more complete simulated picture.

A continuous dialogue between an analytic model approach and actual experiments analysis is crucial when attempting to understand the physical nature of the problem at hand. Discrepancies between the simulated and experimental results tend to originate from assumptions made in the design of the model. This idea facilitates the identification of the origins of these features.

The advantages of proposed spherical coordinates (magnitude and direction) were exhibited. The spherical coordinates enhance results which are random in direction but consistent in overall vector magnitude. Faults that primarily alter the direction of a vibration may require the more traditional Cartesian approach, or an analysis of the directional component of the spherical coordinates.

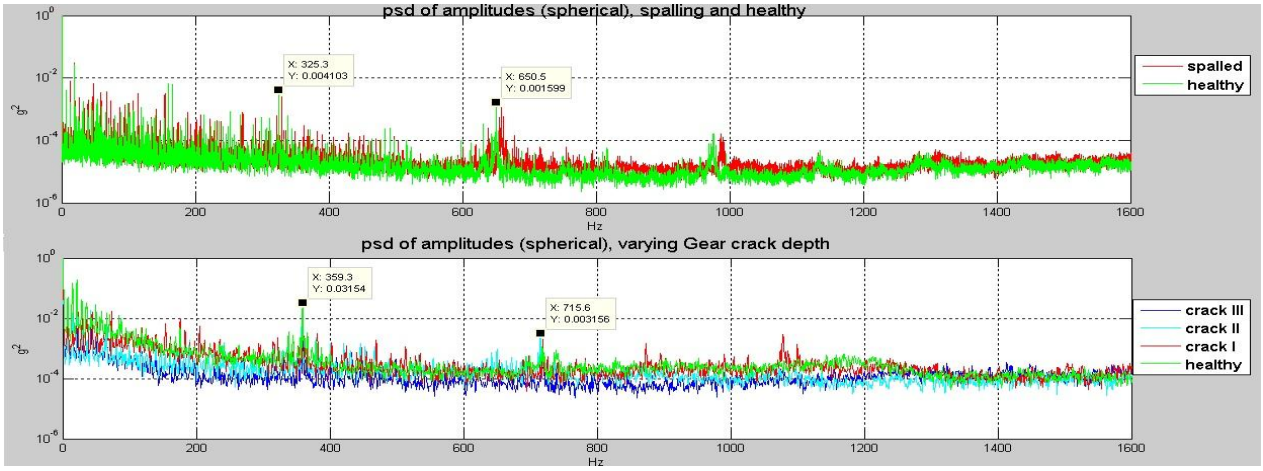


Figure 16. PSD (spherical) of frequency domain, (a) spall vs. healthy, (b) various degrees of cracked gear

NOMENCLATURE

GM	energy summation of the gear mesh feature
h	harmonic index
$\hat{x}$	Fourier transform of x
z	number of teeth on gearwheel
f_s	shaft frequency
df	frequency\order resolution
AM	energy summation of the amp. modulation feature
n	sideband index
FM	energy summation of the freq. modulation feature
$\bar{a}$	acceleration vector in the time domain
a_x	Tangential component of the acceleration vector
a_y	Radial component of the acceleration vector
a_z	Axial component of the acceleration vector

REFERENCES

Dempsey, P. et al., (2007). Investigation of Current Methods to Identify Helicopter Gear Health. NASA, Glenn Research Center, Cleveland, Ohio, USA.

Lewicky, D. et al., (2010). Gear Fault Detection Effectiveness as Applied to Tooth Surface Pitting Fatigue Damage, *Gear Technology Nov./Dec. 2010* pp.48-59, Randall Publications LLC

Klein, R.,(2012). Condition Indicators for Gears. R.K. Diagnostics, *Annual Conference of Prognostics and Health Management Society 2012*, Gilon, Israel.

Klein, R.(2013). Comparison of Methods for Separating Excitation Sources in Rotating Machinery. R.K. Diagnostics, Gilon, Israel.

Dadon, I., Bliznyuk, A., Klein, R., & Bortman, J. (2014). Towards a reliable non-linear dynamic model of damaged gear transmission. *The Eleventh International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, June 10-12, 2014 Manchester, UK.

Chaari, F., Baccar, W., Abbes, M. S., & Haddar, M. (2008). Effect of spalling or tooth breakage on gearmesh stiffness and dynamic response of a one-stage spur gear transmission. *European Journal of Mechanics-A/Solids*, Vol 27, pp 691-705.

Chen, Z., & Shao, Y. (2011). Dynamic simulation of spur gear with tooth root crack propagating along tooth width and crack depth. *Engineering Failure Analysis*, Vol 18, pp 2149-2164.

Chopra, A. K. (2001). *Dynamics of Structures*. USA: Prentice Hall.

Characterization of Fault Size in Bearings

Matan Mendelovich [1], Yitschak Sanders [1], Gideon Kogan [1], Mor Battat [1], Dr. Renata Klein [2], and Prof. Jacob Bortman [1]

[1] *Laboratory for Health, Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer Sheva 84105, Israel*

matan.mendelovich@gmail.com
jitschaks@gmail.com
ggkogan@gmail.com
morbat@post.bgu.ac.il
jacbort@gmail.com

[2]*R.K. Diagnostics, Gilon, P.O.B. 101, D.N. Misgav 20103, Israel*
Renata.Klein@RKDiagnostics.co.il

ABSTRACT

Bearings are important components in rotating machines. An initial small damage in the bearing may cause a fast degradation, which may lead to the machine breakdown. The health condition of bearings can be monitored using proven vibro-acoustic methods effective for detecting bearing faults. However, the existing bearing health indicators do not provide a reliable estimation of the fault characteristics, such as fault size and fault location. As a result, the ability to assess the severity of the bearing damage and to make maintenance decisions is limited.

The presented study is a part of an ongoing research on bearing prognostics, aimed to improve the understanding of the effects of fault size on the bearing dynamics. The research methodology combines dynamic modeling of the faulty bearing with experimental validation and confirmation of model simulations.

In the presented study, small faults (starting from 0.3 mm), simulating incipient damage are generated at increasing sizes by an electrical discharge machine. The recorded vibration data is then analyzed and compared to the vibration signatures predicted by the model. The experimental and the simulation results add new insights on the manifestation of the size of the fault and possible indicators of the damage severity.

1. INTRODUCTION

The ability to assess the bearing condition and to estimate its remaining useful life (RUL) is a key factor for machinery prognostics.

Matan Mendelovich et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Our study is focused on estimating the position and size of the fault, based on the vibration analysis of the bearing. This study continues a former research that was conducted in BGU PHM lab aimed to find indications in the vibration signature of the size of the fault (Kogan, Shaharabany, Itzhak, Bortman & Klein, 2013). In the current study, we seeded groove shaped faults, of width between 0.3 and 1.2 mm into the outer-race of the bearing, which simulates realistic faults that often can be found in damaged bearings.

A 3D dynamic model (Kogan, Bortman, Kushnirsky, & Klein, 2012) was used to simulate faults and to study the effects of fault size and location on the vibration signatures. The analysis of the simulations results supported the interpretation of the experimental results.

This research was done in continuation to previous studies in order to improve the fault size estimation (Elforjani & Mba, (2010) and Sawalhi & Randall, (2011)).

2. EXPERIMENT DESCRIPTION

The experimental system includes two subsystems: a generic test rig (as shown in Fig.1) and a measurement unit. The generic test rig includes an AC motor, one shaft with two flywheels on it, mounted on two bearings.

The measurement unit includes a data acquisition system that is connected to an optic sensor and an accelerometer, the optic sensor measures the rotating speed of the shaft, and the accelerometer measures the vibration signals in three directions and is placed on the tested bearing housing (the right bearing in Fig 1).

Each test run was started with shaft alignment. The shaft speed was measured using Keyence optic sensor and vibrations were measured using a Dytran 3263A2 tri-axial

accelerometer. The data was acquired during 60 seconds with a sample rate of 25000 $\left[\frac{sample}{sec}\right]$.

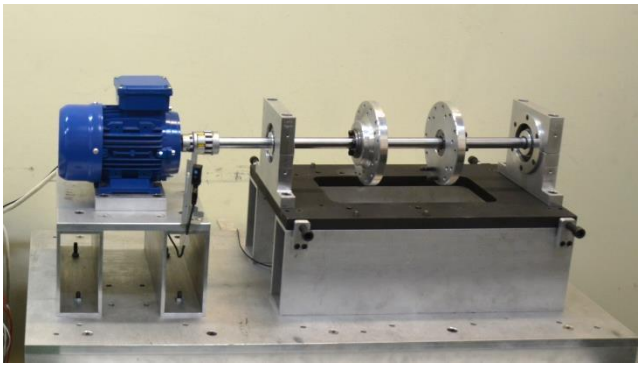


Figure 1. The test rig.

Six different bearings were monitored, a healthy bearing and five others with different fault sizes. Each bearing has been monitored in two fault locations, inside and outside the loading zone. The different fault locations were monitored in order to learn about the influence of the load applied on the fault.

2.1. Test configuration notation

The code of test runs includes 2 variables - the bearing number (noting the fault size) and the location of the fault. For example – "4B" is a bearing with a 0.61[mm] fault size, located 90° to the center of its loading zone. Table 2 summarizes the bearing parameters.

Table 1. Test configuration notation

Bearing number	Fault size [mm]	Location A loading zone	Location B $\mathbf{90^0}$ to the center of the loading zone
1	0	1A	-
2	0.31	2A	2B
3	0.39	3A	3B
4	0.61	4A	4B
5	0.78	5A	5B
6	1.12	6A	6B

Table 2. Bearing properties

Inner diameter	40[mm]
Outer diameter	80[mm]
Width	18[mm]
No. of balls	9
FTF	0.4X
BSF	2.4X
BPFO	3.6X
BPFI	5.4X

3. FAULT GENERATION PROCESS

In order to analyze the vibration signature of faulty bearings, we decided to use bearings that can be disassembled and reassembled without damaging any of the bearing parts during the process.

SKF ETN9 bearings having a "snap" type cage (see Figure 2) that can be removed from the bearing without damage.

Figure 2: SKF ETN9 "snap" type cage

After the bearing disassembly a fault was introduced in the outer ring using an EDM machine with customized copper electrodes (see Figure 3). The faults are thin groove shaped in various widths (see table 1).

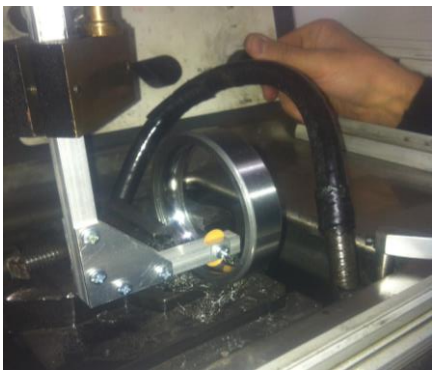


Figure 3: Bearing's outer race in the EDM process

Figure 4: Outer race with a groove shaped fault

4. MODEL DESCRIPTION

A 3D dynamic ball bearing model was developed to study the effect of faults on the bearing dynamic behavior. The aim of the model is to calculate the dynamic response of a bearing with a wide spectrum of faults. The algorithm was implemented numerically in MATLAB.

The dynamics, for each bearings component, are based on the classical dynamic equations

$$\Sigma\overrightarrow{F_f} + \Sigma\overrightarrow{F_n} = m\vec{a},\ \Sigma\left(\vec{R} \times \overrightarrow{F_f}\right) = \boldsymbol{I}\dot{\vec{\omega}}_{xyz} + \vec{\Omega} \times (\boldsymbol{I}\vec{\omega}) \quad (1)$$

where F_f, F_n are respectively the friction and the normal forces that act on a body, with mass m and acceleration $\vec{a}$; and $\Sigma\left(\vec{R} \times \overrightarrow{F_f}\right)$ is the total moment of force acting on a body with a moment of inertia tensor $\boldsymbol{I}$, angular velocity $\vec{\omega}$; body system xyz, with angular velocity $\vec{\Omega}$; and rotational acceleration, within the body system, $\dot{\vec{\omega}}_{xyz}$.

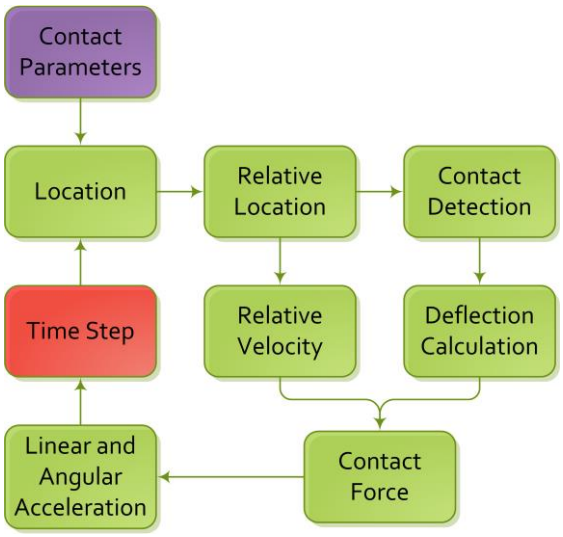


Figure 5. Simplified model algorithm

The relative velocity equation

$$\overrightarrow{v_b} = \overrightarrow{v_a} + \vec{\omega} \times \overrightarrow{ab} \quad (2)$$

Where $\overrightarrow{v_x}$ is the velocity of the body at $\vec{x}$ and $\vec{\omega}$ is the angular velocity of $\overrightarrow{ab}$.

The presented equations describe the motion of all the modeled bodies and are solved using time steps (see Fig. 5). In each time step, the solution of the equations is based on the previous time step solution, assuming a constant acceleration.

The dynamic model was validated by comparison to analytical solutions and known bearing response to local defects (Kogan, Bortman, Kushnirsky & Klein, 2012).

5. DATA ANALYSIS

The analysis process is similar for both the data acquired from the experiments and from the model, and it is described in figure 6. The resampled data rate is 2048 samples/cycle. The envelope of the acceleration data is calculated without filtering and the order domain of the envelope is achieved by calculating the 'Power Spectral Density' of the data, using 32 frames, which gives a resolution of 0.0156 order.

Figure 6: Data analysis process for the experiments and for the model data.

Since the model is simulating the acceleration on the bearings outer race without considering the transmission function to the sensor, we compared the order representation of the simulated data envelope and the order representation of the experimental data envelope. The order of the envelope reflects mainly the effects of the bearing filtering out most of the irrelevant data such as the transmission function and the effects of other rotating components.

Due to the simplicity of the test rig the bearings are the source for the vast majority of the peaks expected in the order of the envelope .Therefor, RMS of the order representation of the envelope is expected to provide a reliable indicator for the fault size. The RMS was calculated up to the 25th order and includes the first six BPFO harmonics and their sidebands.

The RMS level of the envelope of each of the runs was calculated. Then mean RMS value of each test configuration was calculated, average of three runs in similar conditions. Consequently, each configuration of the system (in each direction) is represented in the relevant graphs by a single mean value RMS.

6. RESULTS

6.1. Envelope spectrum

The fault pattern in the envelope spectrum is expected to contain peaks at the ball pass frequency over the outer race (BPFO) as well as lower sidebands caused by modulation of the shaft speed (McFadden & Smith, (1984)). Sidebands are expected due to imperfections of the test system such as unbalance and misalignment. Therefore, unbalance and normal radial clearance where simulated in the model.

The pattern was confirmed in the envelope order spectra of the test runs with faulty bearings and in the order spectra of the model. An example of an order spectrum of the envelope of bearing 1A (fault size 0.31 mm in the loading zone) is shown in Figures 7. Figure 8 contains the envelope order spectrum for the same fault size and location, as analyzed from the vibration signature of the experiment. In both

figures the BPFO, at order 3.6, and its harmonics are dominant, and numerous sidebands corresponding to the shaft speed can also be observed. It should be noted that the sidebands are lower by 2 degrees of order compared to the BPFO harmonics.

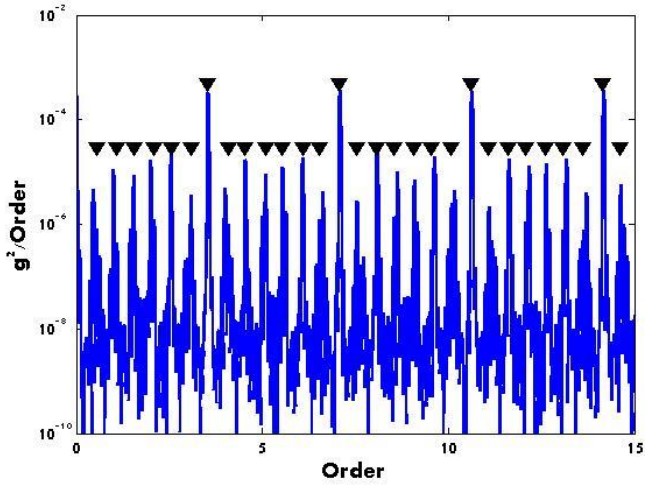


Figure 7. Model based results: order representation of the envelope of bearing with a 0.31mm fault located at the center of the loading zone. The triangles mark the BPFO harmonics and the related sidebands.

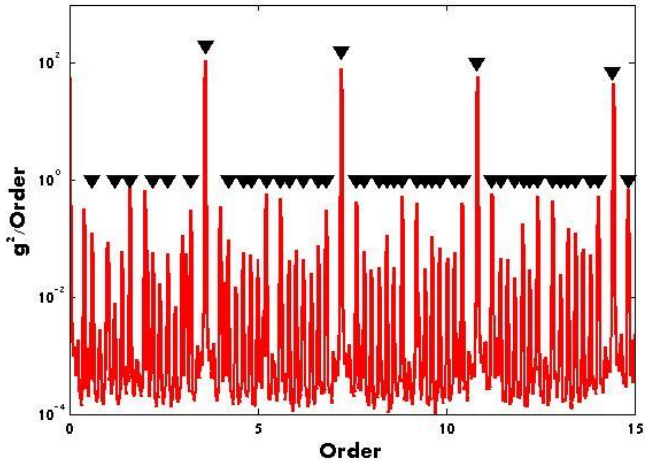


Figure 8. Experimental results: order representation of the envelope of bearing with a 0.31mm fault located at the center of the loading zone. The triangles mark the BPFO harmonics and the related sidebands.

It is notable that both the data from the experiments and from the model have the same general pattern.

6.2. Fault size and location – model

The model results, RMS levels of the envelope up to the 25th order as a function of fault size, are displayed in Figures 9 and 10. As can be seen in Figure 9, the RMS level of the envelope increases with the fault size. The RMS levels in the vertical direction of a fault located in the loading zone is significantly higher compared to the horizontal acceleration because the impulse generated by a ball passing the faulty surface is in the vertical direction. When the fault is located at 90^o to the center of the loading zone small forces (Sawalhi & Randall, (2008)) are applied in the vertical direction and the RMS levels remain constant.

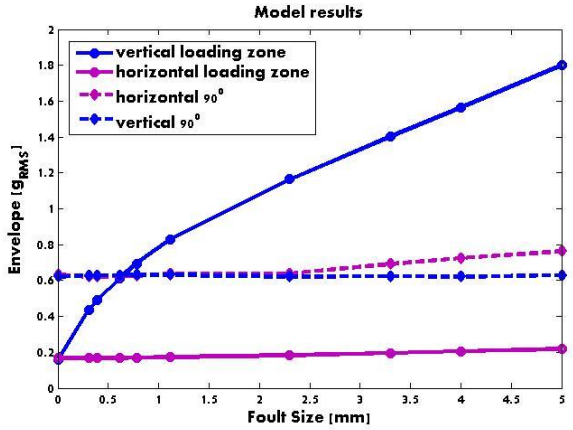


Figure 9. Model based results: RMS levels of envelope acceleration as a function of the fault size.

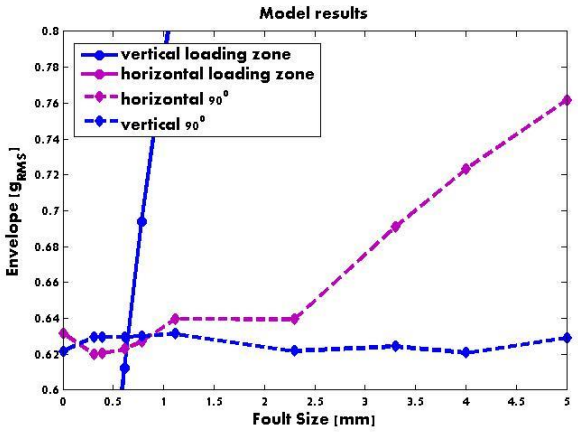


Figure 10. Model based results: RMS levels of envelope acceleration magnified to emphasize the prediction of a fault located at 90°.

It can also be observed that when the fault is located at 90^o , the RMS levels for faults above 2mm, in the horizontal direction are higher than the RMS levels in the vertical direction. The same conclusion was found in a former research (Kogan, Shaharabany, Itzhak, Bortman & Klein, 2013). The behavior of RMS levels for faults located at 90°, as predicted by the model, can be better observed in Figure 10, magnified in the appropriate range. For small size faults, up to 0.78mm the RMS levels are in the same range, whereas for bearings with larger faults, the RMS levels in the

horizontal direction are indeed getting higher compared to the RMS levels in the vertical direction (see Figure 10).

According to the results, it seems that the method proposed in Kogan, et al, 2013, which suggests to use the ratio between the horizontal and the vertical RMS as an indicator of the fault location is not applicable for small faults.

Since the model calculates the accelerations at the location of the fault in the outer race, the RMS levels differ for bearings without faults. In general, the levels of the vibrations at the different locations are not comparable.

6.3. Fault size and location – experimental results

The experimental results, RMS levels of the envelope up to the 25th order as a function of fault size, are displayed in Figures 11 and 12. In general, the trend of RMS level of the envelope corresponds to the defect size both in the horizontal and vertical directions. In addition, as seen in Figure 11, when the fault is located at the center of the loading zone, the envelope RMS levels in the vertical direction are higher than the RMS levels in the horizontal direction, as predicted by the model.

When the fault is located outside the loading zone at 90^o, the RMS levels in the horizontal direction are in the same range as in the vertical direction (see Figure 12). It can also be noted that the RMS levels in the horizontal direction are slightly lower than the RMS levels in the vertical direction, except for the bearing with fault size 1.12mm, as predicted by the model.

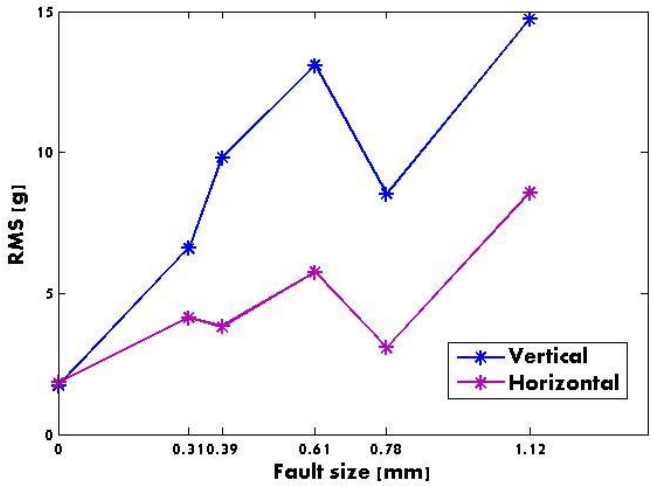


Figure 11. Experimental based results: RMS levels of envelope acceleration as a function of the fault size when fault located at the center of the loading zone (location "A").

When the fault is at the center of the loading zone, the runs with fault size 0.78mm seems to be out of the general trend. It was found that the background level of the relevant tests was extremely low compared to the other tests (up to three decades lower). The reason for this might be the initial alignment of the test kit, since shaft modulation is a result of unbalance and misalignment. Another explanation for the difference in the background level might be a slightly different structure of this particular bearing compared to the other defected bearings.

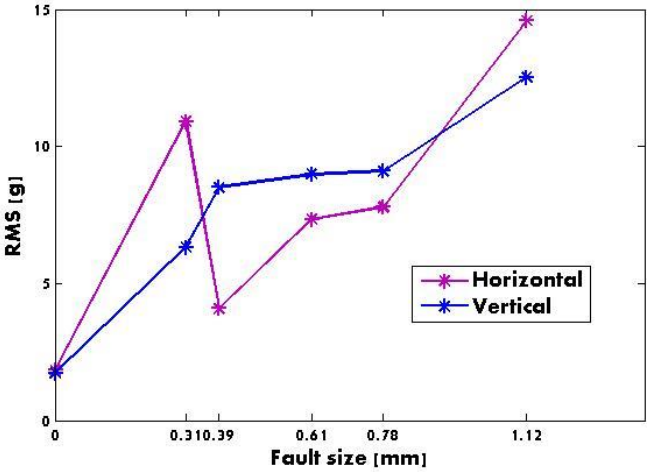


Figure 12. Experimental results of acceleration RMS vs fault size when fault located 90^o to the loading zone (location "B").

6.4. Experiments and model results comparison

The general pattern of the RMS levels as a function of fault size and location is similar in the model and the experiments. Generally, the RMS levels increase as the fault size increases, and the relations between the horizontal and the vertical RMS levels are similar.

RMS levels of the experimental results for a fault in and out of the loading zone were in the same range. However, the model shows a big difference in the RMS levels between the two locations. The reason for the difference is the transfer function from the fault to the sensor, which is not taken into consideration in the model simulations. Moreover, in the model the RMS levels represent the acceleration at two different locations on the outer race. In the experiments, both fault locations are measured at the same location on the bearing housing. The transmission paths from the two locations on the outer race to the sensor differ in the ranges of vibration levels.

7. CONCLUSIONS

A 3D ball bearing dynamical model was compared to test rig experiments in several fault sizes and locations. It was found that the behavior of the acceleration RMS levels as a function of the fault size are similar in the experimental and the model results. In both cases, the general RMS level increases. In addition, a new insight was found about the relation between

the vertical and the horizontal vibration levels as function of fault size and fault location.

It was found that the model provides a good prediction about trends and pattern of localized faults. This fact allows us continue the study of the effects of fault size and location using the model.

NOMENCLATURE

F	Force
I	Moment of inertia
R	Location vector
a	Acceleration
m	Mass
$\mathbf{v}$	Velocity
Ω	Body system angular velocity
$\boldsymbol{\omega}$	Angular velocity

REFERENCES

Sawalhi, N., & Randall, R. B. (2008). *Semi-automated bearing diagnostics – three case studies.* School of Mechanical and Manufacturing Engineering. The University of New South Wales, Sydney, Australia.

McFadden, P. D., & Smith, J. D. (1984). *Vibration Monitoring of rolling element bearing by the high-frequency resonance technique - a review*, Tribology international, Vol. 17, pp 3-10.

M. Elforjani, D. Mba. (2010). *Accelerated natural fault diagnosis in slow speed bearings with Acoustic Emission*. Engineering Fracture Mechanics 77 (2010) 112–127.

N. Sawalhi, R.B. Randall. (2011). *Vibration response of spalled rolling element bearings: Observations, simulations and signal processing techniques to track the spall size*. Mechanical Systems and Signal Processing 25 (2011) 846–870

Kogan, G., Bortman, J., Kushnirsky, A., & Klein, R. (2012). *Ball bearing modeling for faults simulation*, Ninth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, no. 1, pp. 1–8.

Kogan G., Shaharabany S., Itzhak I., Bortman J. & Klein R., (2013). *Towards Model Based Prognostics - Characterization of Fault Size in Bearings*, Annual Conference of the PHM Society 2013.

A Vibration-Based Approach for Stator Winding Fault Diagnosis of Induction Motors: Application of Envelope Analysis

Chao Jin[1], Agusmian P. Ompusunggu[2], Zongchang Liu[1], Hossein D. Ardakani[1], Fredrik Petré[2], and Jay Lee[1]

[1]*NSF I/UCRC Center for Intelligent Maintenance Systems (IMS), Cincinnati, OH, 45221,USA*

jinco@mail.uc.edu
liuzc@mail.uc.edu
davarihn@mail.uc.edu
jay.lee@uc.edu

[2]*Flanders' Mechatronics Technology Centre (FMTC), Heverlee, 3001, Belgium*

agusmian.ompusunggu@fmtc.be
frederik.petre@fmtc.be

ABSTRACT

Induction motors are usually considered as one of the key components in various applications. To maintain the availability of induction motors, it calls for a reliable condition monitoring and prognostics strategy. Among the common induction motor faults, stator winding faults are usually diagnosed with current and voltage signals. However, if the same performance can be achieved, the use of vibration signal is favorable because the winding fault diagnostic method can be integrated with bearing fault diagnostic method which has been successfully proven with vibration signal. Existing work concerning vibration for winding faults often takes it either as auxiliary to magnetic flux, or is not able to detect the winding faults unless severity is already quite significant. This paper proposes a winding fault diagnostic method based on vibration signals measured on the mechanical structure of an induction motor. In order to identify the signature of faults, time synchronous averaging was firstly applied on the raw vibration signals to remove discrete frequency components originating from the dynamics of the shaft and/or gears, and the spectral kurtosis filtering was subsequently applied on the residual signal to emphasize the impulsiveness. For the purpose of enhancing the residual signal in practice, a demodulation technique was implemented with the help of kurtogram. A series of experiments have been conducted on a three-phase induction motor test bed, where stator inter-turn faults can be easily simulated at different loads, speeds and severity levels. The experimental results show that the proposed method was able to detect inter-turn faults in the induction motor, even when the fault is incipient.

Chao Jin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Three-phase induction motors play a vital role in many engineering areas such as high-speed trains, electric vehicles, industrial robots, and machine tools, etc. Unexpected failures of induction motors occurring in these machines can thus lead to excessive downtime and large losses in terms of maintenance cost and lost revenue. Condition-based maintenance (CBM) and predictive maintenance (PdM) have been proven to be a maintenance strategy that can reduce unscheduled downtime and maintenance cost. In CBM, one does not schedule maintenance activities for machines merely according to history of maintenance records and fixed maintenance rules, but also based on the prediction of machine health conditions from sensor data, so that the waste owing to redundant maintenance and failures will be avoided. Such maintenance strategy requires the technologies of: (a) on-line condition monitoring, (b) fault detection and diagnosis, and (c) prognostics.

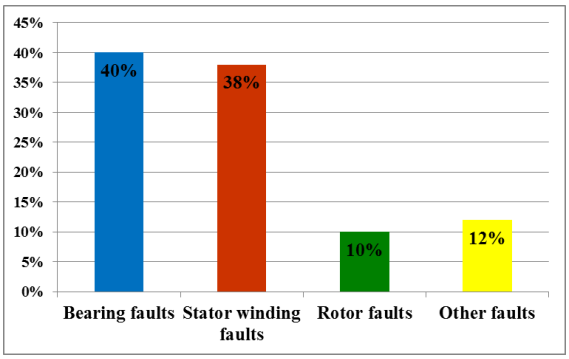


Figure 1. Statistics of failure modes in induction motors

Figure 1 shows the statistical distribution of common failure modes typically occurring in induction motors. Rolling-element bearing and stator winding failures due to insulation degradation contributes to 80% of the causes for unexpected breakdown in induction motors (Jover Rodríguez & Arkkio, 2008). Condition monitoring, diagnosis, and prognostics for rolling-element bearings have been well studied during the past four decades due to its wide applications in almost all the rotary machinery. Vibration-based and motor current signature analysis (MCSA) based monitoring methods for roller-element bearings in induction motors have been widely published in literature. However, the condition monitoring for winding insulation faults, especially vibration-based diagnosis and prognosis methods remain limited.

Winding faults due to insulation degradation can be classified into four types (Ukil, Chen and Andenna, 2011), namely (a) inter-turn short of the same phase, (b) short between coils of same phase, (c) short between two phases, and (d) short between phase to earth. Among them, inter-turn fault is considered to be the most challenging winding fault to be detected in induction motors. The online condition monitoring methods for motor winding faults are summarized in Figure 2. Most of the online monitoring methods are based on current and voltage signals, among which the symmetric component current balance monitoring (Furfari & Brittain, 2002; Eftekhari, Moallem, Sadri and Hsieh, 2013), negative sequence impedance detector (Kliman, Premerlani, Koegl and Hoeweler, 1996), voltage mismatch (Sottile, Trutt and Kohler, 2000; Trutt, Sottile and Kohler, 2002), and Parks vector (Cardoso, 1997) are the most widely referred methods. Nevertheless, these methods require measuring 3-phase high voltage signal from induction motors, which requires expensive sensors and DAQ hardware. Moreover, direct measurements of 3-phase voltages from motor windings are not feasible for online application, and the voltage measurements from the frequency-inverter drive are usually pulse-width modulation (PWM) signals that need additional signal processing process.

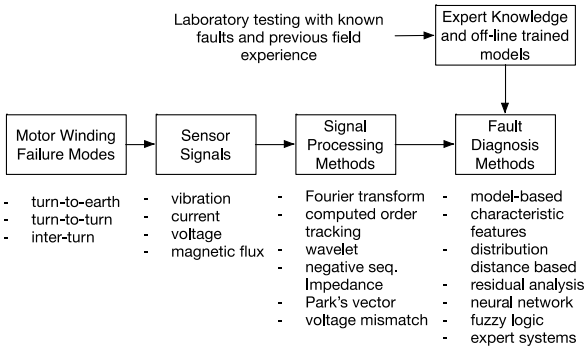


Figure 2. Online condition monitoring methods for motor winding fault (Sin, Soong and Ertugrul, 2003)

Compared with the current and voltage-based winding fault monitoring, vibration-based methods have the advantages of (a) requiring less expensive sensors, (b) requiring less channels for the DAQ system, and (c) monitoring mechanical failures at the same time. Yet vibration analysis for motor winding fault detection has received modest attention due to claimed lower sensitivity. To remedy this gap, this paper proposes a combination of different signal processing techniques to mine and amplify the motor winding fault related features. Time synchronous averaging, spectral kurtosis filtering, and envelope analysis are implemented in the signal processing process. As will be discussed in the results section, the first order of envelope spectrum showed monotonically increasing trend as the level of winding insulation degradation increase.

The remaining part of the paper will be organized as follows: Section 2 discusses the methodology development and theoretical background of the signal processing techniques applied to the motor vibration signals; Section 3 briefly discusses the experimental setup and the test procedure for data generation; Section 4 demonstrates the effectiveness of the proposed vibration signal processing methods and the selected features through the experimental data analysis; and Section 5 summarizes the important findings obtained in this study.

2. METHODOLOGY DEVELOPMENT

2.1. Overall Method

Vibration signal has long been adopted for the diagnosis of mechanical wear in rotary machinery, such as bearings and gearboxes (Randall & Antoni, 2011). One of the elementary assumptions of vibration analysis for rotary machinery mechanical faults is that the concerned fault leads to impulses in vibration signals, which do not occur in the healthy state. Detection of the impulses hidden in the smearing and noise requires advanced signal processing techniques to emphasize the impulsiveness, especially when the fault is incipient. Similar to mechanical faults, induction motor winding faults will generate additional magnetomotive force that is usually reflected in the vibration signal at harmonics of slot frequency and supply frequency (Lamim Filho, Pederiva and Brito, 2014). However, these characteristics are only significant when the faulty turns are around 5% of total windings (Lamim, Brito, Silva and Pederiva, 2013), making it difficult to detect winding faults at an early stage.

Inspired by bearing fault diagnosis, this paper addresses the issue when the inter-turn faults are still preliminary by adopting advanced signal processing tools. As shown in Figure 3, the first step of signal processing was to check the vibration data quality (Jabłoński, Barszcz and Bielecka, 2011; Jablonski & Barszcz, 2013) to guarantee raw data integrity and justify the correctness in the following

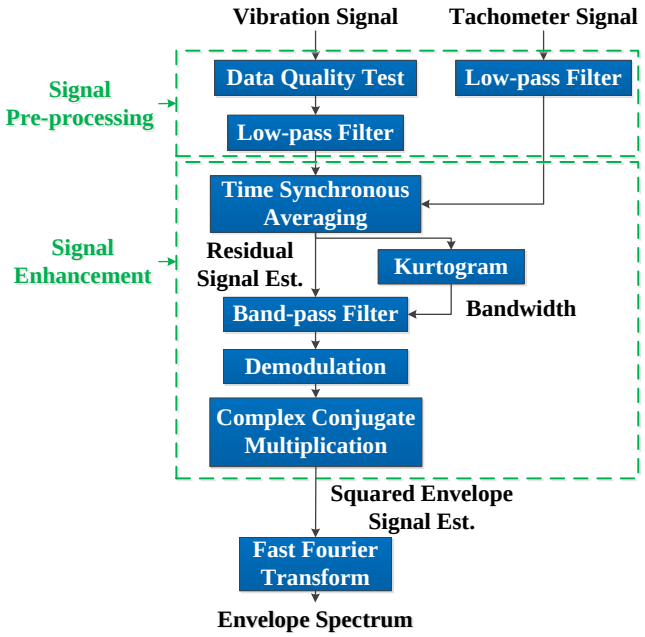


Figure 3. Flowchart of inter-turn fault detection for three-phase induction motors using vibration signal.

analysis. Then, the "corrected" vibration signal and the tachometer signal passed through a low-pass filter to exclude the high frequency noise. The cut-off frequency was set to be one fourth of sampling frequency (in this case 12800 Hz) for the vibration signal, and 10 Hz for the tachometer signal, since the ratio of tachometer is 1/4. After the aforementioned pre-processing steps, time synchronous averaging (TSA) was performed to eliminate discrete frequency component noise (Randall & Antoni, 2011). Then the resonance frequency section of the obtained residual signal estimate with TSA that contained faulty characteristics was enhanced by envelope analysis, whose bandwidth was selected using kurtogram.

The following sub-sections focus on introducing the theoretical background of the tools utilized and explaining why they are effective in detecting inter-turn faults in induction motors.

2.2. Theoretical Background

Instead of going through the calculation of magnetic forces, the induction motor winding fault detection strategy is formulated from the perspective of vibration signal processing. To state mathematically, the problem is to detect the inter-turn faulty signal $x(t)$ buried in the noise $\eta(t)$. And the actual raw signal $s(t)$ we get is the combination of the two, which is (Antoni & Randall, 2006)

$$s(t) = x(t) + \eta(t) \tag{1}$$

Under this problem statement, the following assumptions for this research are proposed:

1. The inter-turn faulty signal $x(t)$ has transients and contains impulses which do not occur or follow a different pattern in the healthy conditions;
2. The noise $\eta(t)$ refers to not only the stationary measurement noise, but also the discrete frequency component, namely the vibration influence of the mechanical parts.

2.2.1. Time synchronous averaging (TSA)

Time synchronous averaging (TSA) is an essential tool for rotating machines that extracts periodic waveforms from noisy data. TSA is performed with respect to a certain shaft according to the tachometer signal as angular position reference. Vibration signals that went through TSA process will have an integer number of orders of the fundamental harmonic (shaft frequency) retained, and other vibration components weakened. If the synchronous-averaged signal is subtracted from the original signal, the residual signal that have the harmonics of the shaft frequency removed will be obtained. Both the synchronous-averaged signal and residual signal contain diagnostic information of different failure mode (Al-Atat, Siegel and Lee, 2011). While there are many different techniques for TSA, zero crossing-based technique is the most widely used.

Zero crossing-based TSA resamples the vibration signal to angular domain where the samples recorded in one shaft rotation are interpolated into a fixed number of data points for each revolution. The number of points per revolution N is derived from Eq. (2):

$$N = 2^{ceiling(\log_2 \max(n))} \tag{2}$$

where n is the number of points between two subsequent zero crossing indices of the tachometer signal (Bechhoefer & Kingsley, 2009).

However, resampling from time domain to angular domain will cause problems for the following signal processing steps since the kernel functions of kurtogram, filtering, and envelope analysis have a constant frequency (Δt) instead of constant angle ($\Delta\vartheta$). Hence the synchronous-averaged signal should be interpolated back to its original time-based sampling mechanism before calculating the residual signal.

The process of obtaining residual signal from TSA is summarized as follows:

(1) Find zero-crossing indices in the tachometer signal and calculate the zero crossing time (ZCT) with interpolation.

(2) For each ZCT, calculate the time between ZTC_k and ZCT_{k+1}, namely, $dZCT_k$, where k is the crossing point index.

(3) Calculate the resampled time interval: dZCT/N, where *N* is given by Eq. (2). Interpolate the signal to the newly resampled time and accumulate the resampled data.

(4) Save the original time stamps for each revolution.

(5) Repeat step (2) through (4) for all the revolutions, and then divide the accumulated N point vector by number of revolutions.

(6) Interpolate the *N* point vector (TSA signal) back to the original time stamps for each revolution, and combine the interpolated TSA signal to get the same length of vector as the original data.

(7) Subtract the combined vector from the original data to get the residual signal.

2.2.2. Spectral kurtosis and kurtogram

Kurtosis as a statistical feature is widely used as a global value to detect the peakiness in a signal. It is defined as

$$k = \frac{E\left[\left(x(t) - E\left(x(t)\right)\right)^4\right]}{E\left[\left(x(t) - E\left[x(t)\right]\right)^2\right]^2} \quad (3)$$

where *E[●]* indicates the averaging calculation. Spectral kurtosis is an extension of kurtosis to a function of frequency, and is known for identifying the impulsiveness in the signal spectrum for rotary machinery fault diagnosis. It is calculated based on the short-time-Fourier-transform (STFT) *X(t,f)* of the original signal. As mentioned by Randall et al in (Randall & Antoni, 2011), spectral kurtosis is defined as

$$K(f) = \frac{E\left[\left(X(t,f) - E\left(X(t,f)\right)\right)^4\right]}{E\left[\left(X(t,f) - E\left[X(t,f)\right]\right)^2\right]^2} - 2 \quad (4)$$

The benefit of spectral kurtosis analysis is that it is able to find the frequency band that contains fault characteristics without requiring a large amount of history data. However, it is then of vital importance that an appropriate window length to be chosen for the STFT. In order to find the optimal window length, or equivalently bandwidth, fast kurtogram was adopted to plot spectral kurtosis against level and frequency. Another task for kurtogram is to find the center frequency with the highest spectral kurtosis value, which is related to the resonance frequency of the motor itself. The incipient vibration winding fault causes will be amplified at this resonance frequency. Reader should be able to observe in Figure 4 that the color in the fast kurtogram indicates the value of kurtosis, and in this particular example the highest kurtosis exists at Level 5.5

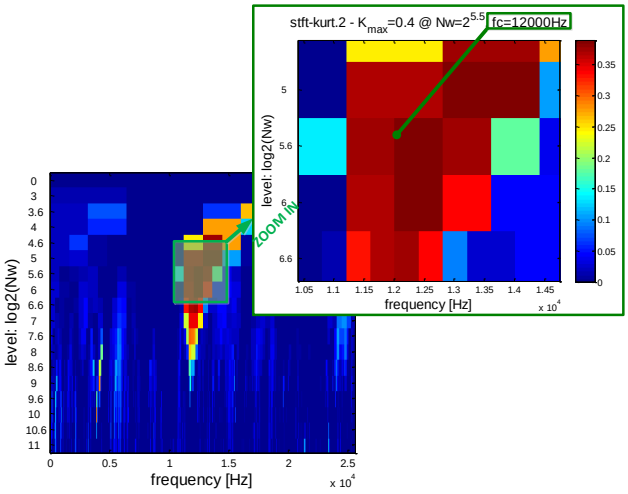


Figure 4. Kutogram of inter-turn fault residual signal at 2000 rpm. The highest kurtosis is 0.4 at Level 5.5 with a center frequency of 12000 Hz.

with a center frequency of 12000 Hz. Even though the fast kurtogram gives the center frequency and the bandwidth, the original power spectrum density still needs to be taken into consideration to finalize the spectrum section that needs to be demodulated later. This part will be shown with graphical explanation in the following sub-section.

2.2.3. Envelope Analysis

Often, the spectrum of raw vibration signal for rotary machinery gives little insight on faulty characteristics due to noise. As mentioned in previous sections, winding faults at early stage induce mechanical impacts that are amplified at the high frequency range of the induction motor system. With kurtogram locating this high frequency range, envelope analysis will further improve the signal to noise ratio and enhance the transients so that the fault can be more easily detected.

The procedure for envelope analysis in this research is described in Figure 5, where the residual signal estimation with TSA is the input and the envelope spectrum is the output. First, a Butter band-pass filter was designed based on the center frequency and bandwidth determined from fast kurtogram. Then the resulting signal was demodulated by following Eq. (5).

$$y(t) = r(t) \times \exp(-j2\pi f_c t) \quad (5)$$

where *r(t)* is the residual signal estimation with TSA, $j = \sqrt{-1}$, f_c is the center frequency, and *y(t)* is the demodulated signal. Afterwards, the demodulated signal went through a low-pass filter with half of the bandwidth as the cutoff frequency. Then the squared envelope signal was calculated by following Eq. (6):

$$e(t) = y(t) \times y^*(t) \quad (6)$$

where *e(t)* represents the squared envelope signal and *y*(t)* represents the complex conjugate of *y(t)*.

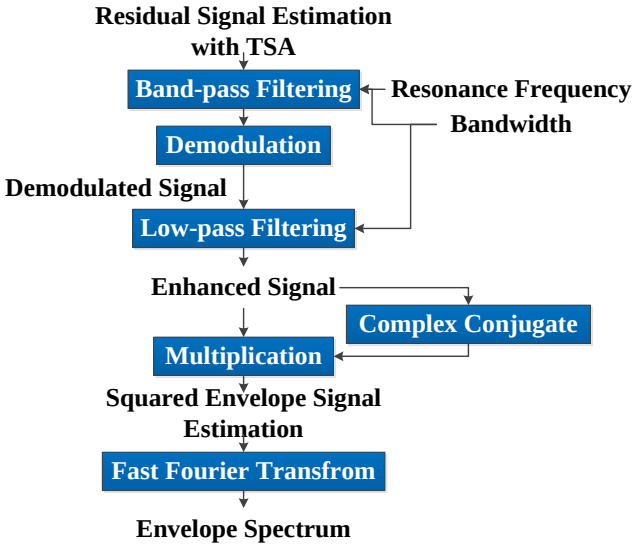


Figure 5. Flowchart of envelope analysis. The resonance frequency (center frequency) and bandwidth are determined with the help of kurtogram.

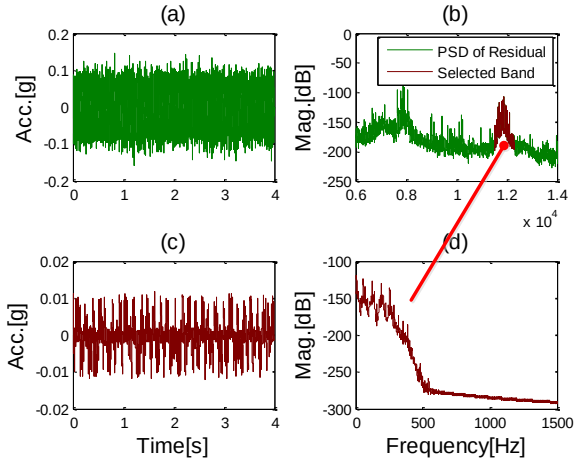


Figure 6. Comparison of time domain and frequency domain signal before and after demodulation: (a) time domain TSA residual signal estimate with kurtosis 3.0459, (b) Welch estimate power spectrum of TSA residual with high-frequency band highlighted in dark red, (c) time domain demodulated TSA residual signal with kurtosis 4.5025, (d) Welch estimate power spectrum density of the demodulated TSA residual. The signal comes from the condition of inter-turn fault. Note that the scales of plots are different.

The result of band-pass filtering and demodulation can be found in Figure 6. In time domain, the emphasis of impulsiveness in the faulty signal is recognized even graphically. Quantitatively, the kurtosis of the signal has increased from 3.1053 to 4.1744. In frequency domain, one can clearly see in Figure 6 (b) that the peaky section centered at approx. 12000 Hz with a bandwidth of 800 Hz is highlighted. This is where the high frequency band that contains the faulty information locates. It was picked up by kurtogram and moved to lower frequency band after demodulation. Discussion on the result of envelope signal and envelope spectrum will be found in Section 4.

3. EXPERIMENTAL SETUP

For conducting this research, a dedicated induction motor test-bed was designed and developed. The test-bed is designed such that one is able to simulate the winding faults with different levels of severity and collect vibration, current, voltage and torque signals from the motor. The winding faults that could be induced in the system include (i) inter-turn and (ii) turn-to-earth faults. The test-bed was also designed to run at different speed regimes and load conditions for multi-regime data collection and analysis. The following sections will briefly describe the test-bed design, the procedure for inducing winding faults and the experiments with different fault conditions.

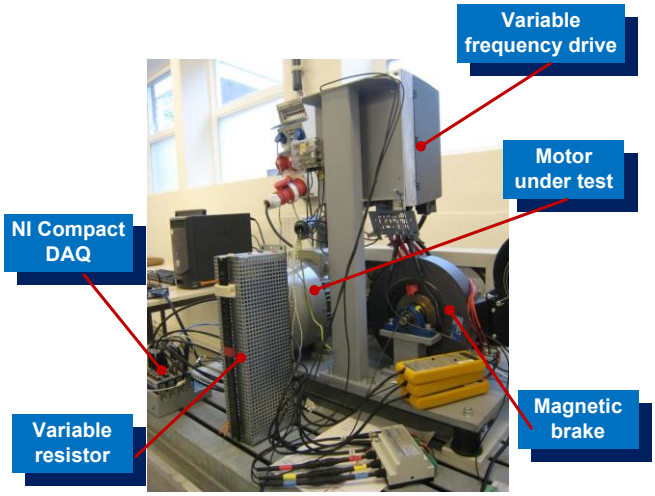


Figure 7. Photograph of the induction motor test bed.

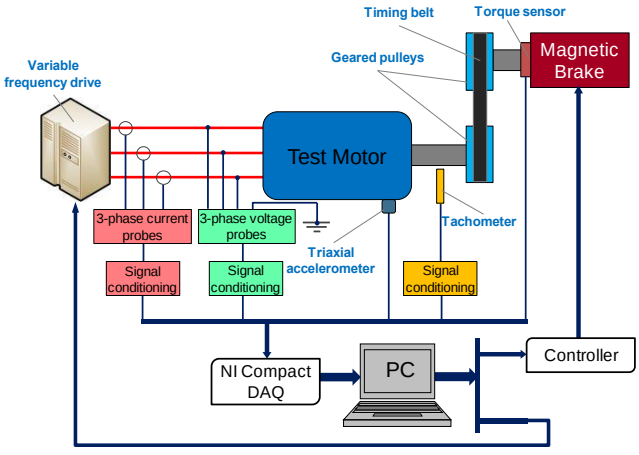


Figure 8. Schematic view of the motor test bed.

3.1. Test Setup

The test-bed consisted of an 11KW, 19.7A, 400V 3-phase induction motor driven by a variable frequency drive (VFD). The rotational speed of the motor could be varied from 0 to 3000 RPM with both stationary and transient modes available. A magnetic brake was connected to the output shaft of the motor through a timing-belt and pulley mechanism. The mechanism allowed the brake shaft to rotate at half of the speed of the motor shaft. By controlling the input current of the brake, an external load varying from 0 to 50 Nm could be applied to the motor. A PC with LabVIEW programs was used to send the control signals to the VFD and magnetic brake controller. A variable resistor with the range of 0-580 Ω was used to simulate different levels of severities in the shorted turns in inter-turn faults. A tri-axial accelerometer was mounted on the top of the housing of the motor to collect the vibration of the motor. A tachometer based on a proximity probe was used to measure the rotational speed of the motor. The head of the tachometer was put towards a 4-tooth flywheel connected to the motor shaft generating 4 pulses per revolution. The experimental setup and the schematic view of the test-bed are shown in Figure 7 and Figure 8.

3.2. Fault Simulation

The winding of the motor used in the test-bed is random-wound (Figure 9). The winding was modified by connecting three shielded wires to the coil of phase w at three locations and the other ends of the wires were brought outside as schematically shown in Figure 10. The inter-turn faults were simulated by connecting the other ends of the wires to a variable resistor. For healthy state simulation, the ends of the three wires were left unconnected. The inter-turn faults were simulated under two different scenarios referred to as inter-turn I and II. In inter-turn I, wires 1 (in orange) and 2 (in green) were connected through a variable resistor. Similarly for inter-turn II, wire 1 was shorted to wire 3 (black) through a variable resistor. By adjusting the resistance to 580 and 300 Ω, two levels of severity for both inter-turn I and II were simulated, as summarized in Table 1.

Table 1. Different fault levels for induction motor

State	Resistance [Ω]	Comment
F1	580	Lowest level
F2	300	Moderate level

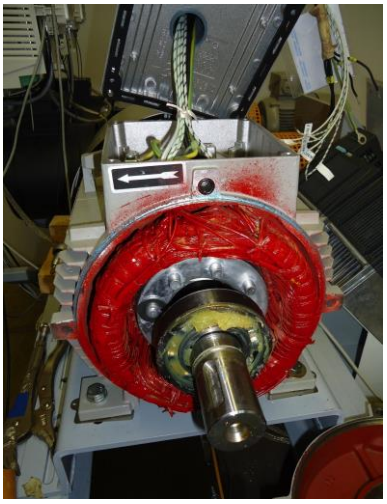


Figure 9. Disassembled motor exposing random would stator winding.

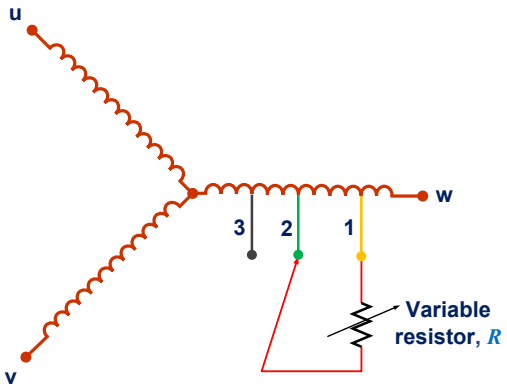


Figure 10. Schematic winding diagram with three taps on the *phase w* winding for different inter-turn fault scenarios.

3.3. Test Procedure

The test was performed at the constant speed of 2000 RPM and constant brake torque of 12 Nm for all the winding conditions. At each level of winding faults, the current i_l flowing through the variable resistor was measured and the corresponding dissipated power P_d was calculated as summarized in Table 2.

Prior to digitizing the signals, each measured signal was passed through a low-pass and an anti-aliasing filter embedded in each channel of the NI data acquisition system. Doing the tests in this way ensures that the potential aliasing problems caused by high frequency noise can be avoided. Depending on the sampling frequency, the cut-off frequency of the anti-aliasing filter was automatically adjusted. The vibration signals were sampled at the rate of 51.2 KHz with the duration of four seconds. The digitized data was stored in the PC and analyzed off-line in MATLAB software.

Table 2. Current and dissipated power through the variable resistor at different states

State	Inter-turn I		Inter-turn II	
	i_l [mA]	P_d [W]	i_l [mA]	P_d [W]
F1	265	40.7	86	4.3
F2	297	26.5	155	7.2

4. RESULTS AND DISCUSSION

Under varying fault severity levels, squared envelope signal estimation was calculated by following the procedure introduced in Section 2.2.3. The result for healthy state, Inter-turn I and Inter-turn II is presented in Figure 11. Compared with the healthy state, it is obvious that the pattern of vibration of the induction motor has changed in time domain for inter-turn fault. The period of one cycle of vibration for the healthy case is approximately 0.0456 s, and the period for both of the inter-turn cases is approximately 0.0300 s, namely 33.3 Hz which is about the same with the rotational speed (2000 RPM/60 s= 33.3 Hz). This is because inter-turn fault has changed the magnetic flux distribution of the induction motor and the faulty characteristic is related to rotating speed. It is also noticeable that the amplitude of the faulty characteristic increases as the fault becomes more severe.

After obtaining the envelope signal, Fourier transform was applied. For the purpose of comparing between different scenarios, amplitudes of the spectrum were normalized according to DC amplitude, which should be the highest; and the frequency domain was also transferred to order domain to help the readers to recognize quickly the feature at the rotational speed. In Figure 12, it is evident that at the first order, inter-turn fault case has a component. And by comparing (3) with (2) in Figure 12, the severity of the fault is also revealed.

Furthermore, a bar plot was generated for all the conditions at different severity levels, which is shown in Figure 13. As one can observe, there is a clear difference between healthy state and inter-turn faults in terms of bar height. In terms of severity, for Inter-turn I and Inter-turn II respectively, amplitudes at F2 in (b) is bigger than those in (a) of Figure 13. Besides, Inter-turn II has a larger value than Inter-turn I, which once again reveals the severity of fault successfully.

Since the values of the order domain amplitudes were normalized between 0 and 1, it can be considered as a metric called hazard value (HV) to quantify inter-turn fault in induction motors. The result is shown in Table 3.

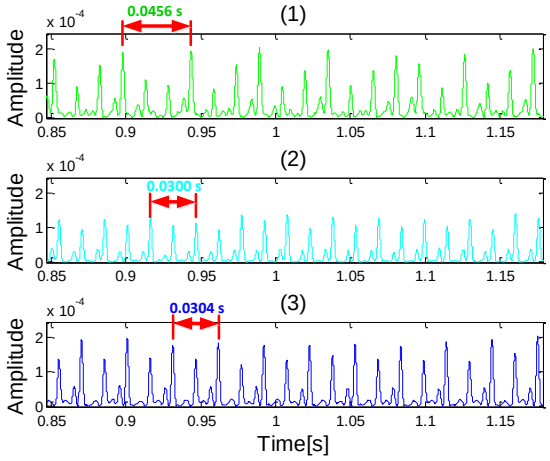


Figure 11. Time domain envelope signals for F1: (1) time domain envelope signal for healthy state with period of approx. 0.0456 s, (2) time domain envelope signal for Inter-Turn I with period of approx. 0.0300 s, (3) time domain envelope signal for Inter-Turn II with period of approx. 0.0304 s. Note that the scales of the three sub-plots are different.

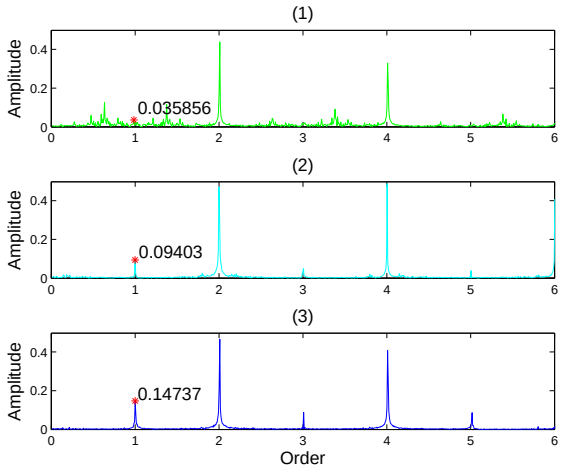


Figure 12. Envelope spectra in order domain for F1: (1) envelope spectrum for healthy state with no harmonic at the first order, (2) envelope spectrum for Inter-turn I with a peak valued at 0.09403 at the first order, (3) envelope spectrum for Inter-turn II with a peak valued at 0.14737 at the first order.

Table 3. Hazard value (HV) of different conditions and severities

Metric	Healthy	Inter-turn I		Inter-turn II	
		F1	F2	F1	F2
HV	0.0359	0.0940	0.2385	0.1474	0.2574

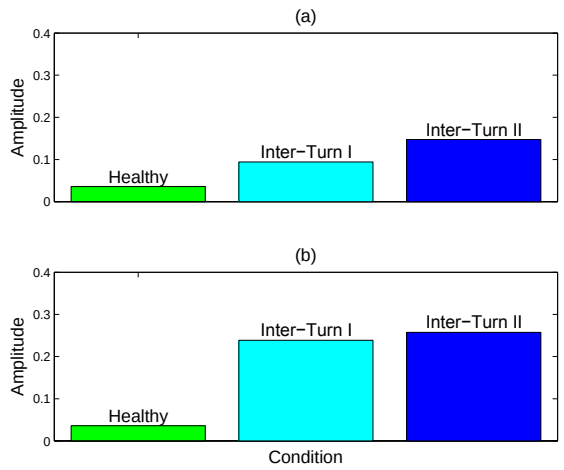


Figure 13. Amplitudes of first order component in envelope spectrum for different conditions and severity levels: (a) amplitudes for all three conditions at severity level F1, (b) amplitudes for all three conditions at severity level F2. The three colors represent healthy state, Inter-turn I, and Inter-turn II, respectively, and they are consistent with previous figures.

5. CONCLUSION

This paper proposes a vibration-based method to detect inter-turn winding fault, which is known to be the hardest to detect even with current and voltage signal. The method was divided into two stages, namely signal pre-processing stage and signal enhancement stage. In the pre-processing stage, data quality check and a low-pass filter were applied on both vibration signal and tachometer signal. In the signal enhancement stage, several techniques were adopted. Time synchronous averaging was used to remove the discrete frequency component noise, and then the residual signal was demodulated at the center frequency and bandwidth selected with the help of kurtogram. The resulting normalized envelope spectrum was converted into order domain, and the component at the first order was able to detect inter-turn fault from the healthy state, and reflect the severity. Note that this method is applied at a constant speed, and time synchronous averaging technique is in fact quite computationally costly. Other techniques to remove the discrete frequency components like cepstrum analysis are to be explored for future work.

REFERENCES

Al-Atat, H., Siegel, D. & Lee, J. (2011). A systematic methodology for gearbox health assessment and fault classification. *Int J Prognostics Health Manage Soc*, vol. 2(1), pp. 16.

Antoni, J. & Randall, R. (2006). The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, vol. 20(2), pp. 308-331.

Bechhoefer, E. & Kingsley, M. (2009). A review of time synchronous average algorithms. *Annual conference of the prognostics and health management society*

Cardoso, A. (1997). The Park's Vector Approach: a general tool for diagnostics of electrical machines, power electronics and adjustable speed drives. *Record of the 1997 IEEE International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, Carry-le-Rouet, France* (261-269)

Eftekhari, M., Moallem, M., Sadri, S. & Hsieh, M.-F. (2013). Online Detection of Induction Motor's Stator Winding Short-Circuit Faults.

Furfari, F. & Brittain, J. (2002). Charles LeGeyt Fortescue and the method of symmetrical components. *Industry Applications Magazine, IEEE*, vol. 8(3), pp. 7-9.

Jablonski, A. & Barszcz, T. (2013). Validation of vibration measurements for heavy duty machinery diagnostics. *Mechanical Systems and Signal Processing*, vol. 38(1), pp. 248-263.

Jabłoński, A., Barszcz, T. & Bielecka, M. (2011). Automatic validation of vibration signals in wind farm distributed monitoring systems. *Measurement*, vol. 44(10), pp. 1954-1967.

Jover Rodríguez, P. V. & Arkkio, A. (2008). Detection of stator winding fault in induction motor using fuzzy logic. *Applied Soft Computing*, vol. 8(2), pp. 1112-1120.

Kliman, G., Premerlani, W., Koegl, R. & Hoeweler, D. (1996). A new approach to on-line turn fault detection in AC motors. *Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS'96., Conference Record of the 1996 IEEE* (687-693)

Lamim Filho, P., Pederiva, R. & Brito, J. (2014). Detection of stator winding faults in induction machines using flux and vibration analysis. *Mechanical Systems and Signal Processing*, vol. 42(1), pp. 377-387.

Lamim, P., Brito, J. N., Silva, V. A. D. & Pederiva, R. (2013). Detection of Electrical Faults in Induction Motors Using Vibration Analysis. *Journal of Quality in Maintenance Engineering*, vol. 19(4), pp. 2-2.

Randall, R. B. & Antoni, J. (2011). Rolling element bearing diagnostics—a tutorial. *Mechanical Systems and Signal Processing*, vol. 25(2), pp. 485-520.

Sin, M. L., Soong, W. L. & Ertugrul, N. (2003). Induction machine on-line condition monitoring and fault diagnosis - a survey. *Australasian Universities Power Engineering Conference* (1-6), Christchurch, New Zealand

Sottile, J., Trutt, F. C. & Kohler, J. L. (2000). Experimental investigation of on-line methods for incipient fault detection [in induction motors]. *Industry Applications Conference* (2682-2687)

Trutt, F. C., Sottile, J. & Kohler, J. L. (2002). Online condition monitoring of induction motors. *Industry Applications, IEEE Transactions on*, vol. 38(6), pp. 1627-1632.

Ukil, A., Chen, S. & Andenna, A. (2011). Detection of stator short circuit faults in three-phase induction motors using motor current zero crossing instants. *Electric Power Systems Research*, vol. 81(4), pp. 1036-1044.

Formal Verification of Complex Systems based on SysML Functional Requirements

Hoda Mehrpouyan[1], Irem Y. Tumer[2], Chris Hoyle[2], Dimitra Giannakopoulou[3], Guillaume Brat[3]

[1] *TSYS School of Computer Science, Columbus State University, Columbus, GA, USA*
mehrpouyan_hoda@columbusstate.edu

[2] *School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR, USA*
irem.tumer@oregonstate.edu, chris.hoyle@oregonstate.edu

[3] *NASA Ames Research Center, Moffett Field, CA, USA*
dimitra.Giannakopoulou@nasa.gov, guillaume.p.brat@nasa.gov

ABSTRACT

As modern systems continue to increase in size and complexity, they pose increasingly significant safety and risk management challenges. A model-based safety approach is an efficient way of coping with the increasing system complexity. It helps better manage the complexity by utilizing reasoning tools that require abstract models to detect failures as early as possible during the design process. This paper develops a methodology for the verification of safety requirements for design of complex engineered systems. The proposed approach combines a *SysML* modeling approach to document and structure safety requirements, and an *assume-guarantee* technique for the formal verification purpose. The assume-guarantee approach, which is based on a compositional and hierarchical reasoning combined with a learning algorithm, is able to simplify complex design verification problems. The objective of the proposed methodology is to integrate safety into early design stages and help the system designers to consider safety implications during conceptual design synthesis, reducing design iterations and cost. The proposed approach is validated on the quad-redundant Electro-Mechanical Actuator (EMA) of a Flight Control Surface (FCS) of an aircraft.

1. INTRODUCTION

In recent years, technological advancements and a growing demand for highly reliable complex engineered systems, e.g., space systems, aircrafts, and nuclear power plants have made the safety assessment of these systems even more important. Moreover, the growing complexity of such systems has made it more challenging to achieve design solutions that satisfy safety and reliability requirements (Wiese & John, 2003; Zio, 2009; N. Leveson, 2011). Hollnagel et al. (Hollnagel, Woods, & Leveson, 2007) recognize the fact that safety violation in complex systems is not necessarily a consequence of components' malfunction or a faulty design. Rather it could be a result of a network of ongoing interactions between all the components and subsystems that introduce undesired behavior. For this reason, Baroth et al. (Baroth et al., 2001) recommends the Prognostic and Health Management System (PHMS) as a new technology to replaces the traditional build-in test (BIT) with intelligent prognostics tools to predict the occurrence of unexpected faults. However, given the local safety properties of each component, it is not a trivial matter to infer the safety and reliability of the whole system (N. G. Leveson, 2009). Well-specified verification formalism and reasoning tools are needed to study the emerging behavior and to perform exhaustive verification of safety properties. A series of safety standards emerged in recent years that recognize this issue and strongly recommended the use of formal verification methods to control the complexity of safety-critical systems, i.e., the international standard on safety related systems (IEC, 1998) and the SAE & EUROCAE standards in the avionic industry (ARP4761, 1996; ARP4754, 1996). However, these standards do not specify how to implement formal approaches throughout the design process.

Strategies for engineered system design emerge from a process of requirement decomposition and transforming requirement models into the conceptual models (Blanchard, 2012; Buede, 2011). Requirement models, noted $\boldsymbol{R}$, capture the design problem being solved and conceptual models, noted $\boldsymbol{S}$, represent the specific solution for the design problem. Therefore, the first step in specifying and formulating a complex system is to capture its requirements $\boldsymbol{R}$ and decompose it into the requirements of its sub-systems and components, noted

Hoda Mehrpouyan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

$R = \{R_1, R_2, ..., R_n\}$. The second step is to create a relationship between design requirements and the system that consists of heterogenous sub-systems, i.e., electrical, mechanical, and software ..., noted $S = \{S_1, S_2, ..., S_m\}$. However, this relationship between the set of design requirements and the set of sub-systems and components is a non bijective relationship. A commonly used formalism to address this problem is to focus on *discrete event system dynamics*. This formulation is extended (Hirtz, Stone, McAdams, Szykman, & Wood, 2002; Nagel, Stone, Hutcheson, McAdams, & Donndelinger, 2008; Kurtoglu & Campbell, 2009) by considering other system features such as structures and functions, so that the predicate ($S_1 \wedge S_2 \wedge ... \wedge S_m \Rightarrow$ Design's Objective) is preserved and satisfied throughout the design process. So the formulation can be summarized as below:

$$\begin{cases} S_i \Rightarrow \{R_k\}_{k\in[1..n]} & S_i \text{ satisfies a sub-set of requirements.} \\ \{S_k\}_{k\in[1..m]} \Rightarrow R_i & R_i \text{ satisfied by sub-set of sub-systems or components.} \end{cases}$$

The process of identifying and proving the correctness of these relationships with regards to design safety requirements is the objective of this paper. The remainder of this paper is structured as follows: section *2* discusses the system oriented approaches and their ability in modeling multi-domain complex engineered system and being exploitable for safety analysis. Furthermore, formal verification methods and the definition of *compositional reasoning* and its commonly used terminologies and operators are introduced as a complementary technique to design requirement analysis. In section *3* an overview of the step-by-step implementation of the compositional reasoning algorithm on the components of the design architectures is explained. Further, section *3* outlines the application of the proposed methodology in the analysis and verification of the safety properties of the quad-redundant Electro Mechanical Actuator (EMA) system design. The paper ends with conclusion.

2. RELATED WORK

Different standards, e.g., (IEEE1220, 2005; ISO-IEC15288, 2002) have defined system design as a multidisciplinary collaborative process that defines, develops, and verifies a system solution which satisfies different stakeholders' expectations and meets public safety and acceptability. Therefore, identification and analysis of the system requirements and designing a system according to the identified requirements are the two inter-correlated and complementary processes of system design. While these standards precisely specify the processes involved in the design of a safety critical systems, Lundteigen et al. (Lundteigen, Rausand, & Utne, 2009) agree that they do not provide methods and tools for efficient design of complex engineered systems. This highlights the need for appropriate methods and tools to support the integration of safety into the design solution.

2.1. SysML for Complex Engineered Systems

Traditional methods and tools used by system engineering are mostly based on a formalism that capture a variety of system features, i.e., requirements engineering, behavioral, functional, and structural modeling, etc. Those with particular focus on requirements engineering are the Unified Modeling Language (UML) (OMG, 2007) to support various aspect of system modeling, Rational Doors (IBM, 2010) to express the requirements, and Reqtify (GeenSys, 2008) to trace the requirements through design and implementation. UML is developed by the Object Management Group (OMG) in cooperation with the International Council of Systems Engineering (INCOSE). UML is an Object-oriented modeling language that allows hierarchical organization of system component models, which in turn results in easier reuse and maintenance of the system model. However, UML was originally developed for software engineers and its primary application is software-oriented; therefore it does not meet all the system engineer's expectations. For example, UML does not provide a notion to represent continuous flows exchanged within the system, i.e., Energy, Material, and Signal (EMS). The analysis of EMS flows are crucial in system design safety verification for identifying the failure propagation path and identifying the common failure modes. For this reason, the SysML profile was developed borrowing a subset of the UML language to meet the requirements of a general purposed language for system engineering.

SysML is an efficient modeling language for constructing models of complex, multidisciplinary, and large-scale systems. SysML enables the designers of a complex system to model the system requirements, structures, behaviors, and parametric values for a more rigorous description of a system under consideration. SysML focuses on the global features of architectural views, whereas other modeling languages such as he Architecture Analysis and Design language (AADL) addresses the more detailed platform-oriented and physical aspects of such systems. Nevertheless, the wide variety of notations provided by SysML lacks formal and detailed semantics required for requirements verification. The goal of this paper is to bridge the gap between semi-formal approaches, e.g., SysML and formal verification methods, e.g., model-checkers to provide the system designers an integrated method to manage and verify the safety properties of complex engineered systems.

2.2. Model Checking and Formal Verification

Model checking is one of the approaches to formal verification of finite state hardware and software systems (Henzinger,

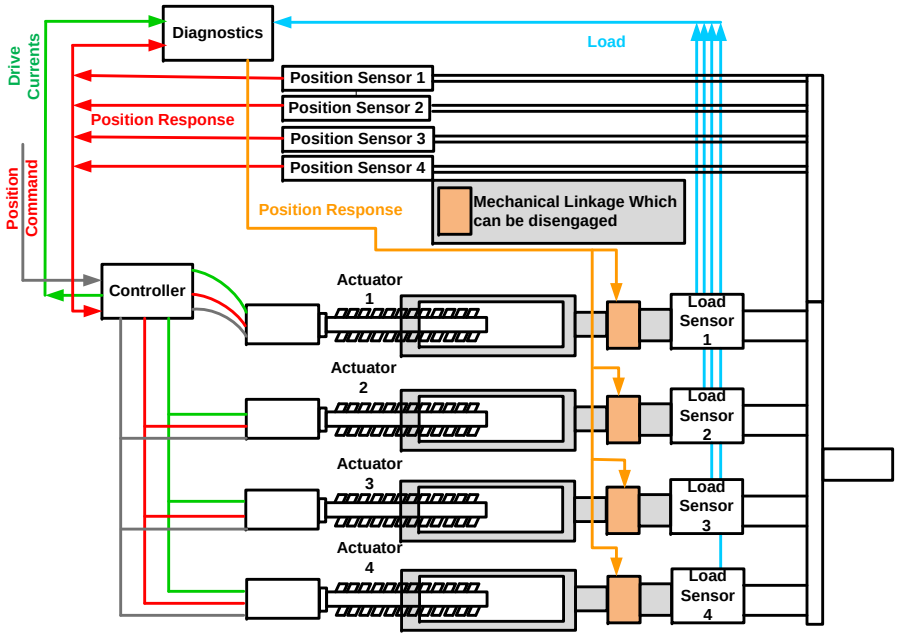


Figure 1. Quad-Redundant EMA Scheme.

Ho, & Wong-Toi, 1997; Henzinger, Nicollin, Sifakis, & Yovine, 1994). In this approach, a design will be modeled as a state transition system with a finite number of states and a set of transitions. The design model is in essence a finite-state machine, and the fact that it is finite makes it possible to execute an exhaustive state-space exploration to prove that the design satisfies its requirements. Since there is an exponential relationship between the number of states in the model and number of components that make up the system, the *compositional reasoning* approach is used to handle the large state-space problem. The compositional reasoning technique decomposes the safety properties of the system into local properties of its components. These local properties are subsequently verified for each component. However, Barragan et al. (Barragan, Roth, Faure, et al., 2006) emphasizes the difficulty of transforming the global system requirements into multi-level sub-system and component's local safety properties that need to be verified by a model checker for the design of large scale complex engineered systems. More specifically, the decomposition of complex engineered systems into multi-domain sub-systems involving electrical, mechanical, and software components makes the refinement and traceability of the global safety properties very difficult. Therefore, a systematic approach is required to acquire abstract requirements along with safety properties, and map them to system components (Evrot, Petin, & Mery, 2006). Following the work of many researchers, it is concluded that the early stages of system design are the most critical in ensuring that the designed system satisfies its safety requirements (Tumer, Stone, & Bell, 2003; Stone, Tumer, & Stock, 2005; Kurtoglu & Tumer, 2008; Tumer & Smidts, 2011), this paper aims at addressing this challenge using the system-oriented SysML-based modeling approach combined with formal verification technique.

2.3. Case Study

As depicted in Fig. 1, a quad-redundant Electro-Mechanical Actuator (EMA) (Balaban et al., 2009) for the Flight Control Surfaces (FCS) of an aircraft, developed in a program sponsored by NASA, is used to illustrate and validate the proposed approach. The positions of the surfaces, A, C, and D, in Fig. 2, are usually controlled using a quad-redundant actuation system. The FCS actuation system responds to position commands sent from the flight crew, B in Fig. 2, to move the aircraft FCS to the command positions.

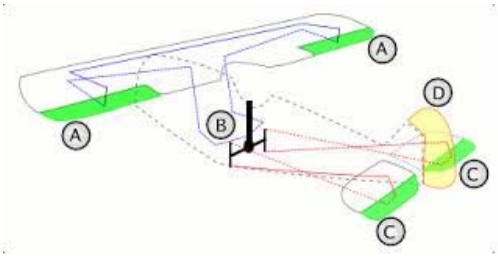


Figure 2. Basic Aircraft Control Surfaces.

The EMAs are arranged in a parallel fashion; therefore, each actuator is required to tolerate a fraction of the overall load. To meet safety requirements, each actuator is required to take on the full expected load from the FCS in the extreme case where all three of the four actuators become non-operational. In addition, the design should also consider other issues such as the possibility of the actuators becoming jammed. If one actuator becomes jammed in this parallel arrangement, it will prevent the other ones from moving. Therefore, a mechanism

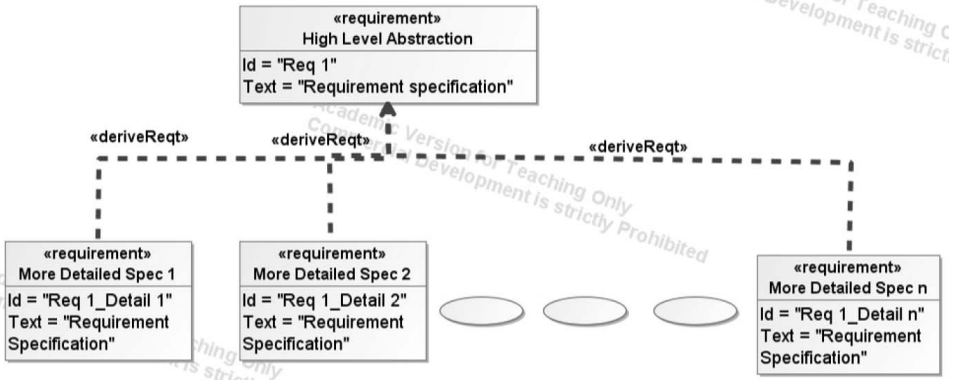


Figure 3. Requirements Decomposition.

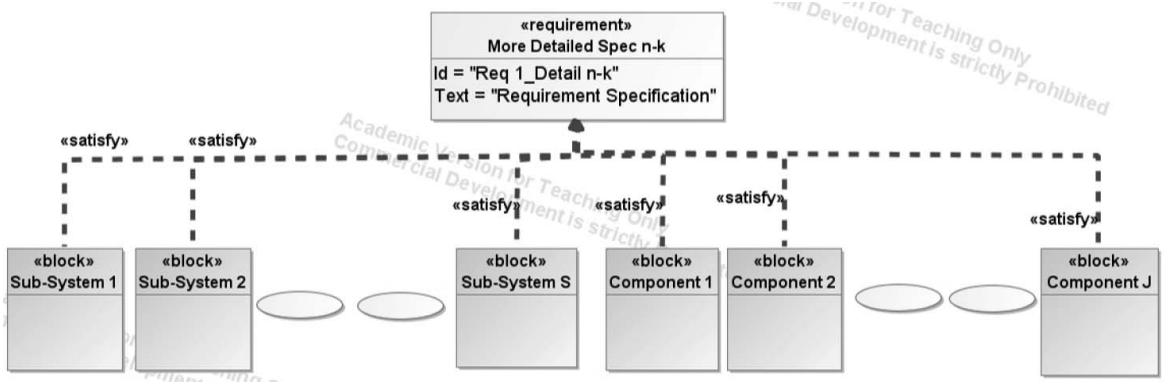


Figure 4. Requirements Mapping.

to disengage faulty actuators from the rest of the system is required to avoid the faulty actuators from becoming deadweights. Once an EMA is disengaged from the system it cannot be re-engaged automatically. It is envisioned that this will happen on the ground, once the aircraft has landed.

In order for the design to be reliable, additional redundancies in other components of the system, such as load and position sensors are required. Thus, a fully quad-redundant scheme is envisioned, as depicted in Fig. 1. As illustrated, the design features redundancy in the EMAs and the sensor feedback signals. The position command is fed to the control loop, while the load from the FCS is shared by the EMAs. The individual load, current, and position response signals from each EMA are used to perform separate diagnostics on each EMA. Therefore, faults are isolated to the individual actuators, which facilitates adaptive on-the-fly decisions on disconnecting degraded EMAs from the load. A dedicated diagnostics block performs actuator health assessments, and makes decisions on whether or not to disengage any faulty actuators from the flight control surface. The disengagement is made possible by mechanical linkages, which can be disconnected from the output shaft coupling.

3. METHODOLOGY

Design requirements are the specification of safety constraints initially defined in the design. Requirements are modeled at different levels of abstractions. For example, a higher level of abstraction is used when expressing the global system properties and a low level of abstraction is used when expressing the required features for each system component, i.e. the barriers and materials to be used. Managing this set of specifications is based on iterative decomposition and substitution of the abstract requirements by the requirements that are more concrete.

3.1. Safety Requirements Modeling Using SysML

A SysML requirement diagram enables the transformation of text-based requirements into the graphical modeling of the requirements which can be related to other modeling elements. Fig. 3 depicts the decomposition of a single abstract requirement into several more explicit ones. A study by Blaise et al. (Blaise, Lhoste, & Ciccotelli, 2003) confirms the effectiveness of such diagrams to facilitate the structuring and management of requirements that are traditionally expressed in natural languages.

The next step in the requirement analysis phase consists of mapping the requirements to the corresponding system components or functions. System components are modeled as

part of the structural design of a system. The structural design model corresponds to the system hierarchy in terms of systems and subsystems, which are modeled using the Block Definition diagram (BDD). SysML blocks are the best modeling elements to model multi-disciplinary systems and are especially effective during system specification and design. They are effective because blocks are not only able to model logical or physical decomposition of a system, they also enable designers to define specification of software, hardware, or human elements.

Fig. 4 illustrates how a single requirement can be satisfied by a set of sub-systems and components. The requirement diagram is connected to the structure diagram by a cross connecting element known as *satisfy*. A requirement can be satisfied by a component or subsystem. Furthermore, the detailed modeling of sub-systems and components are possible through the use of Internal Block Diagram (IBD). In addition, blocks are a reusable form of description that can be applied throughout the construction of system modeling if necessary. Another advantage of using blocks during the design process is their ability to include both structural and behavioral features, such as properties and operations that represent the state of the system and behavior that the system may display.

Including properties as part of the requirement modeling is specifically important when verifying safety requirements. As Madni. (Madni, 2007) demonstrated, safety is a changing characteristic of complex systems that, once integrated into the design, is not preserved unless enforced throughout system operation. Hollnagel et al. (Hollnagel et al., 2007) also confirms that safety is a feature that results from what a system does, rather than a characteristic that the system has. Therefore, the proof of safety is provided by the absence of failures and accidents. For this reason, "safety-proofing" a system design is never absolute or complete. Consequently, the proposed approach does not guarantee safe system operation, instead provides formal proof that certain very specific behavioral parameters will be achieved. It is for this reason that in this paper safety is viewed as a system property.

A complete proof of safety is possible through a formal definition of different properties that are linked to each high-level abstract and low-level detailed requirements. Fig. 5 represents how a requirement, property, block, and behavioral model are connected to one another. For example, *allocate* as a cross connecting principle in SysML is used to connect a behavior to a component in a structure diagram.

In the proposed approach, individual components' behavior in the system are modeled as Labeled Transition Systems (LTSs), LTSs basically represent a finite state system. The properties of the LTSs make it ideal for expressing the behavioral model of system components. The LTS model is expressed graphically, or by its alphabet, transition relation, and

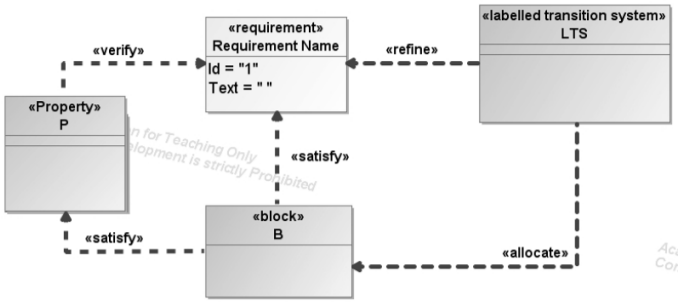


Figure 5. Requirements Traceability.

states including single initial state. The LTS of the system is constructed from the LTS of its subsystems, and is verified against safety properties of the design requirements (Fig. 5).

3.2. Safety Requirements Verification

A model-based verification approach is proposed based on the *behavioral* models of design components, where behavioral specifications are associated with each component. These specifications are then used to analyze the overall design architecture. In this approach, a design will be modeled as a state transition system with a finite number of states and a set of transitions. The design model is in essence a finite-state machine, and the fact that it is finite makes it possible to execute an exhaustive state-space exploration to prove that the design satisfies its requirements. Since there is an exponential relationship between the number of states in the model and number of components that make up the system, the *compositional reasoning* approach is used to handle the large state-space problem. The compositional reasoning technique decomposes the safety properties of the system into local properties of its components. These local properties are subsequently verified for each component. The combination of these simpler and more specific verifications guarantees the satisfaction of the global safety of the overall system architecture design. It is important to note that, the safety requirements of the components are satisfied only when explicit assumptions are made on their environment. Therefore an *assume-guarantee* (Cobleigh, Giannakopoulou, & Păsăreanu, 2003; Giannakopoulou, Păsăreanu, & Barringer, 2005; Nam & Alur, 2006; Chaki, Clarke, Sinha, & Thati, 2005) approach is utilized to model each component with regards to its interaction with its environment, i.e, the rest of the system and outside world.

Since, the LTSs are based on graphical modeling, they can easily become unmanageable for large complex systems. Therefore, an algebraic notation known as Finite State Process (FSP) (Rodrigues, 2000) is used to define the behavior of processes in a design. FSP is a specification language as opposed to a modeling language, with semantics defined in terms of LTSs. Every FSP model has a corresponding LTS description and vice versa. An example FSP and LTS model of the *Electro Mechanical Actuator* (EMA) unit of the quad-redundant

EMA of Fig. 1 is provided in Table 1 and Fig. 6 respectively.

Table 1. FSP Description of EMA

```
1 : EMA = (recLoad → ApplyLoad → (allLoadsCompleted → EMA
2 :          | jam → block → Jammed)),
3 : Jammed = (recLoad → Jammed
4 :          | disengage → unblock → Disengaged),
5 : Disengaged = (recLoad, allLoadsCompleted, timeout → Disengaged).
```

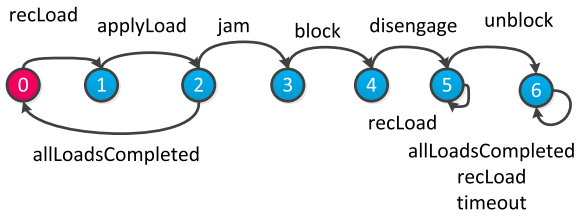


Figure 6. LTS Model of the EMA Subsystem.

In the defined model, a EMA receives the load command from the controller and carries out the operation. The Electro Mechanical Actuator is modeled in Table 6 with *Jammed* and *Disengaged* as part of its definition. If during the time of maintaining the specified torque or load the EMA functions according to specification, the signal *"all loads are completed"* is sent to the controller. Otherwise, the EMA is considered non-operational or jammed. In the jammmed mode, the EMA is incapable of maintaining the required load and prevents the rest of the EMAs from moving. Therefore, it needs to be disengaged from the system.

After system modeling, the actual analysis of the models is carried out utilizing the Assume Guarantee Reasoning (AGR) verification technique. In the assume-guarantee methodology, a formula contains a triple $\langle A\rangle\ M\ \langle P\rangle$, where M is defined as a component, P is a safety property, and A is an assumption or constraint on M's environment. The formula is proven correct if whenever M is a component within a system satisfying A, then the system also guarantees P.

The simplest assume guarantee rule for checking a safety property P on a system with two components M_1 and M_2 can be defined as following (Henzinger, Qadeer, & Rajamani, 1998; Chaki et al., 2005):

Rule ASYM

$$\frac{\begin{array}{ll}1: & \langle A\rangle\ M_1\ \langle P\rangle \\ 2: & \langle \mathit{true}\rangle\ M_2\ \langle A\rangle\end{array}}{\langle \mathit{true}\rangle\ M_1 \parallel M_2\ \langle P\rangle}$$

The first rule is checked to ensure that the generated assumption restricts the environment of component M_1 to satisfy P. For example, the assumption A is that there is no Electromagnetic Interference (EMI) or Radio Frequency Interference (RFI) in the environment where component M_1 operates; hence, P is satisfied. The second rule ensures that component M_2 respects the generated assumption. For example, M_2 will not generate any EMI and RFI while operating. If both rules hold then it is concluded that the composition of both components also satisfies property P ($\langle \mathit{true}\rangle\ M_1 \parallel M_2\ \langle P\rangle$).

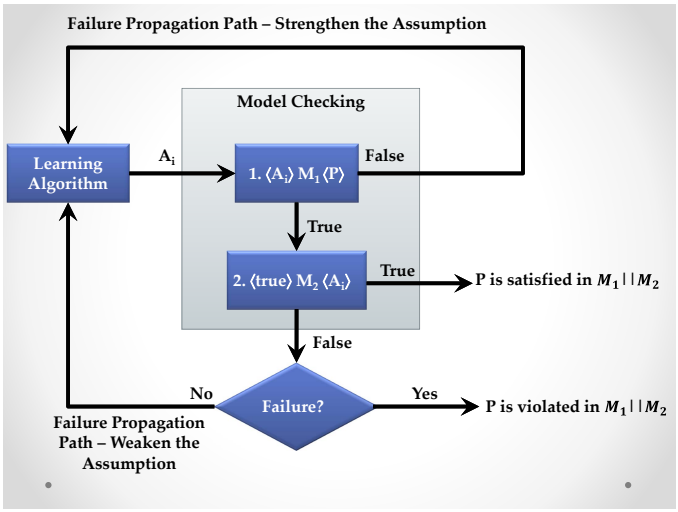


Figure 7. An Overview of the Algorithm that Generates Assumptions.

In this research, the algorithm in (Giannakopoulou, Pasareanu, & Cobleigh, 2004) is used to automatically generate assume-guarantee reasoning at the component, subsystem, and system level. The objective is to automatically generate assumptions for components and their compositions, so that the assume-guarantee rule is derived in an incremental manner. The framework of Figure 7 depicts the steps involved in performing automated assume-guarantee reasoning while generating the assumptions. If rule (1) is violated, it means that the assumption is too weak, so it does not prevent M_1 from reaching its failure state. Based on the generated failure propagation path, the algorithm *learns* a new assumption with more restriction on the environment which makes the assumption stronger than the previous one. The iteration continues until the first rule of $\langle A\rangle\ M_1\ \langle P\rangle$ is addressed. The next step is to check the second rule $\langle \mathit{true}\rangle\ M_2\ \langle A\rangle$. If the rule still holds, then it is concluded that $\langle \mathit{true}\rangle\ M_1 \parallel M_2\ \langle P\rangle$. If the check fails, the algorithm performs analysis on the returned failure propagate path to determine the reason for the failure. If the analysis reveals that A is not the weakest assumption, i.e., elimination of both EMI and RFI is not necessary and only the elimination of EMI suffices to satisfy P, then the learning algorithm will generate a new assumption. If the rules are not satisfied with the generated assumptions, it is concluded that $\langle \mathit{true}\rangle\ M_1 \parallel M_2\ \langle P\rangle$ violates the property P.

4. APPLICATION ON THE CASE STUDY

In the case study of Fig. 2, the Flight Control Surface (FCS) must meet rigorous safety and availability requirements be-

Table 2. Requirement Mapping.

Requirement	Component(s)
Safety Requirement 1	quad-redundant EMAs
Safety Requirement 1.2	quad-redundant EMAs
Safety Requirement 1.2.1	Diagnostics
Safety Requirement 1.2.2	EMAs
Safety Requirement 1.2.3	Controller, Position Sensor, and Shaft

fore it can be certified. The FCS has two types of dependability requirements:

- *Integrity*: the FCSs must address safety issues such as loss-of control resulting from aircraft system failures, or environment disturbances.
- *Availability*: the system must have a high level of availability.

Therefore, it is critical for the FCS to continue operation without degradation following a single failure, and to fail safe or fail operative in the event of a related subsequent failure. The movement of the FCS is controlled by a quad-redundant EMAs. A block diagram of the quad-redundant EMAs is depicted in Fig. 8. As seen from the figure, the model consists of an EMA block which is an hierarchical representation of four independent EMAs. Each EMA is modeled via the Internal Block Definition diagram (IBD). The individual EMA legs receive the common position command, but act independently of each other and share the flight control surface load among themselves.

Fig. 9 depicts a set of high-level requirements. To facilitate the verification process, each level of requirements are associated with a formal FSP using property stereotype in SysML. Therefore, satisfying a property *P1* is the same as satisfying properties *P1.1, P1.2, and P1.3*.

The next phase consists of identifying the design architecture (Fig. 8), including sub-systems and components to map each requirement to a traceable source. As depicted in Fig. 4, requirements mapping are made possible by using the *satisfy* relationship to link a single or set of blocks to one or more requirements. The requirements mapping of quad-redundant EMAs is presented in Table. 2.
In order to transform the requirements and the design architecture presented in Fig. 8 into a finite model, we use FSP. As an example, consider the following FSP model of a *controller subsystem* of the quad-redundant EMAs: The controller gets the load command from the command unit and actively regulates the current to each EMA at every time step. The difference between the external load and the total actuator load response is used to accelerate or decelerate the output shaft. If the controller perceives that the output shaft position response is falling behind the commanded position, it will increase the current flow to the EMAs. As depicted in Table 3, in the FSP description of the controller, a repetitive behavior is defined using a recursion. In this context, recursion is recognized as a behavior of a process that is defined in terms of itself, in order to express repetition.

Table 3. FSP Description of Controller

```
1: Controller = (getLoad[l:L] → Controller[l]),
2: Controller[t:L] = (timeout → Controller
3:       | sendLoad→allLoadsCompleted→getShaftPosition[x:Positions]
4:       →if (x ≥ t) then (missionComplete→Controller)
5:       else Controller[t]).
```

The partial LTS model of the controller is depicted in Fig. 10. The *controller* performs action $<getLoad[l..4]>$, and then behaves as described by $<Controller[l]>$. $Controller[l]$ is a process whose behavior offers a choice, expressed by the choice operator "|". $Controller[l]$ initially engages in either $<timeout>$ or $<SendLoad>$. The action $<timeout>$ is performed when all actuators fail, otherwise $<SendLoad>$ is utilized. Subsequently, after sending the required load to each EMA, feedback signals are sent to inform the controller of completion of tasks by labeling the action with *<all Loads Completed>*. This results in the controller to perform the action *<get Shaft Position>*. At this stage, the controller compares the new position with the required shaft position, if the shaft has reached the required position then the *<mission is completed>*. Otherwise, the behavior is repeated until the shaft reaches the required position.

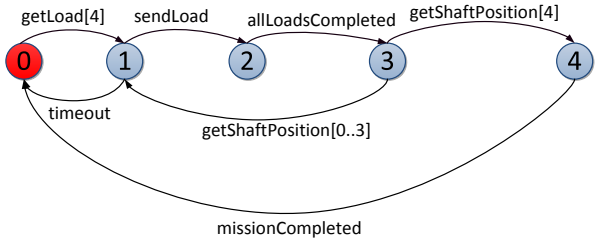


Figure 10. LTS Model of the Controller Subsystem.

After modeling the behavior of each component and sub-system, the design is described by a *composition* expression. In the context of system design engineering, the term composition is similar to the coupled model. The coupled model defines how to couple several component models together to form a new model, similarly, composition groups together individual state machines. Such an expression is called a parallel composition, denoted by "||". The "||" is a binary operator that accepts two LTSs as an input argument. In the joint behavior of the two LTSs, the transition can be performed by any of the LTS if the action that labels the transition is not shared with the other LTS. Shared actions have to be performed concurrently. Table 4 depicts the FSP of the joint behavior of *EMA* and *controller*. The composed LTS model of the two subsystems consists of 161 states and 62 transitions. The shared action between the two models is the $<sendLoad>$ action from the controller and the $<recLoad>$ action from

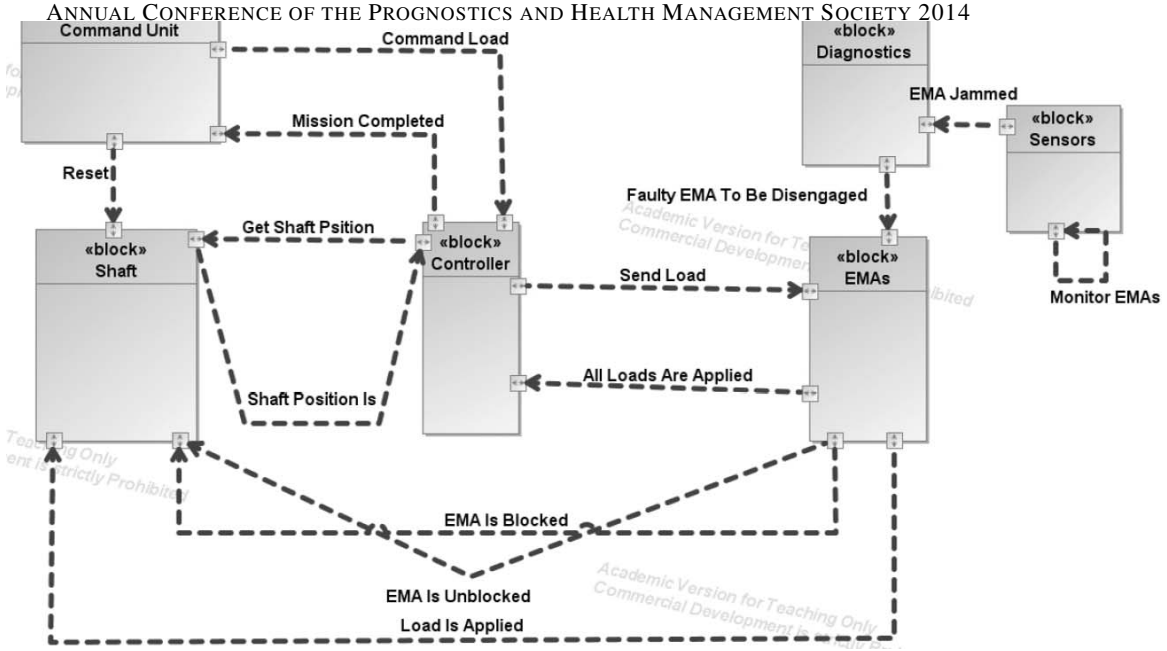


Figure 8. Structural Model of the Quad-redundant EMAs.

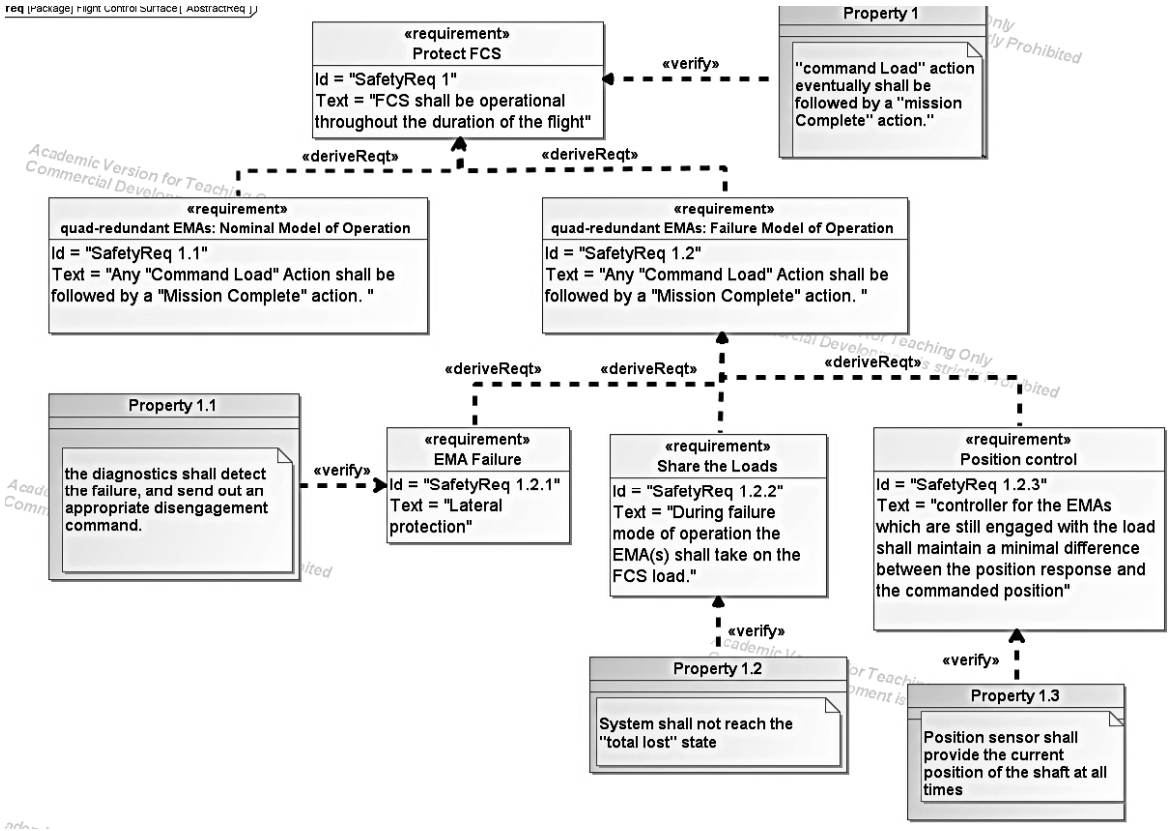


Figure 9. Quad-redundant EMAs High-Level Requirements.

the EMA, therefore, these two are required to be performed synchronously. In order to change action labels of an LTS, the *relabeling* operator "/" is used, e.g., { recLoad / sendLoad }.

Table 5 presents some of the state transitions (or sequence of actions) produced by the composed model. Two possible executions under the EMA's nominal and faulty conditions are considered. In nominal mode, the EMA receives a request from a controller to provide two unit loads. At each

Table 4. Parallel Composition of EMA (Table 1) and Controller (Table 3)

1: || **Leg = (EMA || Controller) / { recLoad / sendLoad }.**

time step, EMA performs one unit load and repeats until the output shaft reaches the required position that is when the $<missionComplete>$ actions is performed. In the failed mode, initial actions are the same as in nominal mode until an EMA jams. The jammed EMA blocks the rest of the system from moving until it is disengaged. The process is followed by the $<Unblock>$ action which unblocks the shaft allowing the rest of the system to be freed. By this time, the EMA has provided one unit load before being disconnected from the rest of the system. Since, the $<ShaftPositionIS>$ shows the current position of the shaft being one instead of two, the EMA is required to perform one more unit of load. However, the disengaged EMA is incapable of doing so resulting in a $<timeout>$. The $<timeout>$ occurs only when there are no EMAs to perform the required load.

Table 5. Leg Subsystem: Two Possible Transitions

EMA: Nominal Mode	EMA: Failure Mode
1: ctrl_getLoad.2	*1*: ctrl_getLoad.2
2: EMA_recLoad	*2*: EMA_recLoad
3: EMA_performLoad	*3*: EMA_performLoad
4: LoadsCompleted	*3*: **EMA_jam**
5: ShaftPositionIs.1	*4*: **Shaft_block**
6: EMA_recLoad	*5*: **EMA_Disengage**
7: EMA_performLoad	*6*: **Shaft_Unblock**
8: LoadsCompleted	*7*: LoadsCompleted
9: getShaftPosition.2	*8*: ShaftPositionIs.1
10: EMA_performLoad	*9*: **timeout**
11: missionComplete	–

So far, we provided the basis for decomposing and modeling the system based on the modular description of the design components and subsystems. In the next phase, the process of expressing the desired safety properties in terms of a state machine or LTS is described. The advantage is that both the design and its requirements are modeled in a syntactically uniform fashion. Therefore, the design can be compared to the requirements to determine whether its behavior conforms to that of the specifications. In the context of this work, the properties of a system are modeled as safety a FPS. A ***safety*** FPS contains no failure states. In modeling and reasoning about complex systems, it is more efficient to define safety properties by directly declaring the desired behavior of a system instead of stating the characteristics of a faulty behavior. In a FSP, the definition of properties is distinguished from those of subsystem and component behaviors with the keyword *property*.

Based on the requirement decomposition model of Fig. 9, the composition model of the properties *P1.1, P1.2, and P1.3* is presented by the following generic (or parameterized) safety property with the following constants and a range definitions is used:

- const N =4 \\ *number of faulty EMAs* [1]
- const M =4 \\ *number of EMAs*
- range EMAs = 1..M \\ *EMA identities*

In order to prevent the system from reaching the catastrophic event of $<timeout>$, it is essential to complete the mission and provide the required loads based on the command signal. The property of Table 6, maintains a count of faulty EMAs with the variable f. To model the fact that every command signal must be followed by a $<missioncomplete>$, property *P1*, the processes in lines 3 and 8 are required to constrain the number of faulty EMAs (f) to a number defined by the parameter of the property (*e.g. N=4*).

Table 6. FSP Model of Safety Property

1: property
2: Fault_Tolerance(N=4) = Jammed[0],
3: Jammed[f: 0..M] =(when(f ≤ N)commandLoad[L] → CompleteMission[f]
4: |when (f>N) commandLoad[L] → Jammed[f]
5: |d[EMAs].jam → Jammed[f+1]
6: |missionComplete → Jammed[f]),
7: CompleteMission[f:0..M] = (missionComplete → Jammed[f]
8: |when (f<N) d[EMAs].jam → CompleteMission[f+1]
9: |when (f= =N) d[EMAs].jam → Jammed[f+1]).

If the above property is predefined with $N = 2$, permitting only two out of four EMAs to fail during the system operation, the verification algorithm of Fig. 7 verifies that the safety property is satisfied.

However, when the property is instantiated allowing four EMAs to fail, the safety analysis verifies that the property is violated and a failure propagation path is produced. Therefore, the generic safety property modeled in Table 6 verifies that the system never reaches the failure condition of *total loss* if and only if *N ≤ M-1* where *N* is the number of faulty EMAs and *M* is the total number of EMAs.

From the result of case study: the characterization of the system architecture by its subsystems and components improves requirements specification, tracking, and modeling. In addition, the FSP annotation of the failure behavior of each of component, and the system level safety analysis based on components' interaction lead to achieving a manageable verification procedure. As the compositional reasoning approach significantly reduces the number states to be explored, exhaustive checking of the entire state space is made feasible without the need for a exhaustive search. This is especially important where the exhaustive simulation is too expensive and non-exhaustive simulation can miss the critical safety violation.

5. CONCLUSION

There is a growing demand for formal methods and tools that facilitate the specification and verification of complex engi-

[1] by default is set to *4* but it can be redefined during the instantiation process.

neered systems design. Also, safety standards for the design of safety-critical systems strongly recommend the use of formal verification approach as part of the certification process. However, these standards do not specify how formal approaches can be implemented. Alternatively, system engineering semi-formal techniques for elicitation and structuring the requirements of complex engineered systems are essential part of the design for electing the conceptual design that satisfies the identified requirements.

In this paper, we have proposed a system modeling and verification approach that combines these apparently contradictory views. The semi-formal SysML techniques based on requirement and block diagrams combined with formal verification methods based on the assume-guarantee reasoning are used to prove that the behavior of sub-systems and components satisfies the design requirements. The proposed approach is based on the mapping between the hierarchical decomposition model of the requirements and properties to be satisfied, functions and behaviors to be realized, and sub-systems and components to be implemented.

The future work will continue in verifying more sophisticated system, while taking into consideration safety properties that are formulated using the temporal operators, i.e., until, before, or after. More complex temporal properties will be tested. In the case of temporal properties, satisfying the system property is not always equivalent to satisfying a local composition of sub-properties. The modified verification algorithm will use linear temporal logic (LTL) as a specification formalism.

REFERENCES

ARP4754, S. (1996). Certification considerations for highly-integrated or complex aircraft systems. *Society of Automotive Engineers Inc*.

ARP4761, S. (1996). Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. *SAE International, December*.

Balaban, E., Saxena, A., Goebel, K., Byington, C., Watson, M., Bharadwaj, S., ... Amin, S. (2009). Experimental Data Collection And Modeling For Nominal And Fault Conditions On Electro-mechanical Actuators. In *Annual conference of the prognostics and health management society* (pp. 1–15).

Baroth, E., Zakrajsek, J., Powers, W., Fox, J., Prosser, B., Pallix, J., & Schweikard, K. (2001). Ivhm (Integrated Vehicle Health Management) techniques for future space vehicles. In *37th joint propulsion conference.*

Barragan, I. S., Roth, M., Faure, J.-M., et al. (2006). Obtaining temporal and timed properties of logic controllers from fault tree analysis. In *Proceedings of the 12th ifac symposium on information control problems in manufacturing, incom 2006, saint-etienne, france.*

Blaise, J.-C., Lhoste, P., & Ciccotelli, J. (2003). Formalisation of normative knowledge for safe design. *Safety Science*, *41*(2), 241–261.

Blanchard, B. S. (2012). *System engineering management* (Vol. 64). Wiley. com.

Buede, D. M. (2011). *The engineering design of systems: Models and methods* (Vol. 55). John Wiley & Sons.

Chaki, S., Clarke, E., Sinha, N., & Thati, P. (2005). Automated Assume-guarantee Reasoning for Simulation Conformance. In *Computer aided verification* (pp. 241–246).

Cobleigh, J. M., Giannakopoulou, D., & Păsăreanu, C. S. (2003). Learning Assumptions For Compositional Verification. In *Tools and algorithms for the construction and analysis of systems* (pp. 331–346). Springer.

Evrot, D., Petin, J.-F., & Mery, D. (2006). Formal specification of safe manufacturing machines using the b method: Application to a mechanical. In *Information control problems in manufacturing* (Vol. 12, pp. 281–286).

GeenSys. (2008). *Reqtify.* www.geensys.com.

Giannakopoulou, D., Păsăreanu, C. S., & Barringer, H. (2005). Component Verification With Automatically Generated Assumptions. *Automated Software Engineering*, *12*(3), 297–320.

Giannakopoulou, D., Pasareanu, C. S., & Cobleigh, J. M. (2004). Assume-guarantee verification of source code with design-level assumptions. In *Proceedings of the 26th international conference on software engineering* (pp. 211–220).

Henzinger, T. A., Ho, P., & Wong-Toi, H. (1997). Hytech: A Model Checker for Hybrid Systems. *Electronics Research Laboratory, College of Engineering, University of California.*.

Henzinger, T. A., Nicollin, X., Sifakis, J., & Yovine, S. (1994). Symbolic Model Checking for Real-time Systems. *Information and Computation*, *111*(2), 193-244.

Henzinger, T. A., Qadeer, S., & Rajamani, S. K. (1998). You assume, we guarantee: Methodology and case studies. In *Computer aided verification* (pp. 440–451).

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A Functional Basis For Engineering Design: Reconciling And Evolving Previous Efforts. *Research in engineering Design*, *13*(2), 65–82.

Hollnagel, E., Woods, D. D., & Leveson, N. (2007). *Resilience engineering: Concepts and precepts*. Ashgate Publishing, Ltd.

IBM. (2010). *Rational doors.* Available from: http://www-01.ibm.com/software/awdtools/doors.

IEC. (1998). 61508 functional safety of electrical/electronic/programmable electronic safety-related systems. *International electrotechnical commission.*

IEEE1220. (2005). *IEEE standard for application and man-*

agement of the systems engineering process. IEEE New York, NY, USA.

ISO-IEC15288. (2002). *Systems engineering system life cycle processes.* International Standardization Organization.

Kurtoglu, T., & Campbell, M. I. (2009). Automated Synthesis Of Electromechanical Design Configurations From Empirical Analysis Of Function To Form Mapping. *Journal of Engineering Design*, *20*(1), 83–104.

Kurtoglu, T., & Tumer, I. Y. (2008). A graph-based fault identification and propagation framework for functional design of complex systems. *Journal of Mechanical Design*, *130*(5), 051401.

Leveson, N. (2011). *Engineering a safer world: Systems thinking applied to safety*. MIT Press.

Leveson, N. G. (2009). The need for new paradigms in safety engineering. In *Safety-critical systems: Problems, process and practice* (pp. 3–20). Springer.

Lundteigen, M. A., Rausand, M., & Utne, I. B. (2009). Integrating rams engineering and management with the safety life cycle of iec 61508. *Reliability Engineering & System Safety*, *94*(12), 1894–1903.

Madni, A. (2007). Designing for resilience. *ISTI Lecture Notes on Advanced Topics in Systems Engineering*.

Nagel, R. L., Stone, R. B., Hutcheson, R. S., McAdams, D. A., & Donndelinger, J. A. (2008). Function Design Framework (FDF): Integrated Process And Function Modeling For Complex Systems. In *Asme 2008 international design engineering technical conferences & computers and information in engineering conference (idetc/cie 2008)* (pp. 273–286).

Nam, W., & Alur, R. (2006). Learning-based Symbolic Assume-guarantee Reasoning With Automatic decomposition. In *Automated technology for verification and analysis* (pp. 170–185). Springer.

OMG, O. (2007). *Unified modeling language (omg uml).* Superstructure.

Rodrigues, R. W. (2000). Formalising UML Activity Diagrams Using Finite State Processes. In *Proc. of the 3rd intl. conf. on the unified modeling language, york, uk.*

Stone, R. B., Tumer, I. Y., & Stock, M. E. (2005). Linking product functionality to historic failures to improve failure analysis in design. *Research in Engineering Design*, *16*(1-2), 96–108.

Tumer, I. Y., & Smidts, C. S. (2011). Integrated design-stage failure analysis of software-driven hardware systems. *Computers, IEEE Transactions on*, *60*(8), 1072–1084.

Tumer, I. Y., Stone, R. B., & Bell, D. G. (2003). Requirements for a failure mode taxonomy for use in conceptual design. In *Proceedings of the international conference on engineering design, iced* (Vol. 3).

Wiese, P. R., & John, P. (2003). *Engineering design in the multi-discipline era: A systems approach*. Wiley.

Zio, E. (2009). Reliability engineering: Old problems and new challenges. *Reliability Engineering & System Safety*, *94*(2), 125–141.

BIOGRAPHIES

Hoda Mehrpouyan Dr. Hoda Mehrpouyan is a Professor in TSYS School of Computer Science at Columbus State University. Her research focuses on model-based systems engineering, resilience and safety analysis, information technology, simulation and verification to support the design of complex systems. She received her Ph.D. from Oregon State University in 2014 and holds a M.S. degree in Software Engineering from Linköping University in 2011. Prior to returnung to academia, She spent 7 years in industry as a system delivery consultant and programmer analyst.

Irem Tumer Dr. Irem Y. Tumer is a Professor in Mechanical, Industrial, and Manufacturing Engineering at Oregon State University, where she leads the Complex Engineered System Design Laboratory, and Associate Dean for Research and Economic Development for the College of Engineering at OSU. Her research focuses on the overall problem of designing highly complex and integrated engineering systems with reduced risk of failures, and developing formal methodologies and approaches for complex system design and analysis. Since moving to Oregon State University in 2006, her funding has largely been through NSF, AFOSR, DARPA, and NASA. Prior to accepting a faculty position at OSU, Dr. Tumer led the Complex Systems Design and Engineering group in the Intelligent Systems Division at NASA Ames Research Center, where she worked from 1998 through 2006 as Research Scientist, Group Lead, and Program Manager. Dr. Tumer has been Conference Chair for ASME Design for Manufacturing and the Lifecycle conference in 2000, Program Chair for IEEE Reliability Society?s Prognostics and Health Management Conference in 2008, and Program Chair (2011) and Conference Chair (2012) for ASME?s International Design Theory and Methodology Conference; and is current Associate Editor for ASME Journal of Mechanical Design and the International Journal of Prognostics and Health Management, and guest editor for AEIDAM journal. She received her Ph.D. in Mechanical Engineering from The University of Texas at Austin in 1998.

Christopher Hoyle Dr. Hoyle's research focuses on decision making in engineering design, with emphasis on the early design phase when uncertainty is high and the potential design space is large. More specifically, he works in the areas of decision-based design (linking consumer preferences and enterprise-level objectives with the engineering design process), uncertainty quantification and management, and complex system design. Areas of technical expertise include uncertainty propagation methodologies, Bayesian statistics and modeling, stochastic consumer choice modeling, optimization, and design automation. Prior to returning to academe, Dr. Hoyle spent 15 years in industry as a project engineer

and engineering manager, concerned primarily with electronics packaging and with managing the trade-offs between performance, manufacturability, and cost.

Dimitra Giannakopoulou Dr. Giannakopoulou is a Research Computer Scientist with the NASA Ames Research Center, and a member of the Robust Software Engineering Group. Her work is concerned with applying modular and compositional formal verification techniques to autonomous systems and architectures. Before joining Ames, she was a Research Associate with the Department of Computing, Imperial College, University of London, UK, working on methods for the specification and automatic verification of distributed systems. She has graduated from the Dept of Computer Engineering and Informatics, University of Patras, Greece. She hold an MSc with distinction from Imperial College, in "Foundations of Advanced Information Technology", and since March 1999, a PhD degree from Imperial College, University of London.

Guillaume Brat Dr. Brat is employed by Carnegie-Mellon University and he conduct research in software verification within the Robust Software Engineering group in the Intelligent Systems Division at NASA Ames. He received an M.Sc. and Ph.D. from the ECE Department at The University of Texas at Austin. He is Principal Systems Scientist at CMU Silicon Valley serving as an IPA at NASA Ames Research Center. He has been the Assistant Area lead for Robust Software Engineering since October 2009. The group conducts research on new verification and validation techniques, mostly based on formal methods.

A Tool Chain for the V&V of NASA Cryogenic Fuel Loading Health Management

Johann Schumann[1], Vanesa Gomez-Gonzalez[2], Nagabhushan Mahadevan[3], Michael Lowry[4], Peter Robinson[5], and Gabor Karsai[6]

[1] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
johann.m.schumann@nasa.gov

[2] *USRA, Mountain View, CA 94043, USA*
vanesa.gomezgonzalez@nasa.gov

[3,6] *Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37212, USA*
nag,gabor@isis.vanderbilt.edu

[4,5] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
{Michael.R.Lowry, Peter.I.Robinson}@nasa.gov

ABSTRACT

Complex machinery like spacecraft, aircraft, or chemical plants are equipped with fault detection and diagnosis systems. Due to their safety-critical nature, such diagnosis systems have to undergo rigorous Verification and Validation (V&V). In this paper, we present a tool suite to facilitate V&V of the deployed diagnostic system. The V&V relies on the paradigms of *cross validation* (to compare the diagnosis results of the deployed reasoner against those of other, more advanced reasoners), *automatic fault scenario generation* (to support extensive testing and coverage analysis), and *parametric model analysis* (to enrich test sets and for robustness and sensitivity analysis). We present the application of this tool architecture towards the V&V of the diagnosis system based on the TEAMS tool suite towards a subsystem in the NASA cryogenic fuel loading facility.

1. INTRODUCTION

Modern complex systems, like the NASA loading facility for cryogenic rocket fuel, are equipped with extensive fault detection and diagnosis systems to quickly detect off-nominal conditions and to diagnose faulty components. For the NASA Kennedy Cryo facility, the commercial TEAMS tool suite (`http://www.teamqsi.com`) is being used for modeling and diagnosis. Obviously, such a plant is highly safety-critical. Thus, fault detection and diagnosis must undergo rigorous V&V in order to ensure that the diagnostic system properly models the physical plant and any associated detectors, so as to minimize the number of false and missing alarms during operation.

In this paper, we present a tool architecture that has been designed to support V&V of TEAMS diagnostic models. Our modular set of tools allows the user to carry out a multitude of V&V use cases and is based upon three basic paradigms: *cross-validation*, *automatic fault scenario generation*, and *parametric analysis*. Our tools are augmented with report generators and a number of advanced statistical analysis and visualization capabilities.

Any diagnostic model is ultimately based on a simplified and abstracted model of the underlying physical plant. TEAMS/RT (Real Time) models are based upon multi-signal diagnosability analysis. Here, the outcome of individual tests ("pass", "fail", or "unknown") results in sets of components (or failure modes) known to be "good", "bad", "suspect", or "unknown", based upon an efficient algorithm using the model's diagnosability matrix (D-matrix). Because of its time-boundedness and efficiency, this kind of discrete diagnosis algorithm has become popular in the aerospace domain, although aspects of timing, fault propagation, fault probabilities, or physical model dynamics cannot be expressed. For real-time applications, the TEAMS/RT diagnosis engine is typically wrapped by custom code for data acquisition, discretization, and filters for noise and transient reduction. The V&V of this wrapper code is as critical as the V&V of the D-matrix. Timing

Johann Schumann et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and fault propagation information, as well as data on component reliability is available in an extended TEAMS Designer model, which is used in an off-line mode for designing instrumentation. This additional information can also be obtained through information provided by subject-matter experts.

Our *cross-validation* tools use such additional information to facilitate deep model analysis. The provided TEAMS models are translated into a different, more expressive modeling paradigm (in our case Timed Failure Propagation Graphs and Bayesian networks) and are enriched with additional information. Then, failure scenarios are executed using reasoners for these paradigms, and results are compared and analyzed.

One of our reasoners is a Timed Failure Propagation Graph (TFPG) reasoner, which uses models that have been generated from the TEAMS models. These TFPG models capture the faults (failure modes), and their propagation effects to trigger one or more anomalies (tests). Additionally, the TFPG models can account for cascading effects of the failures, mode and timing constraints in the failure propagation, and additional information such as failure rate expressed in terms of Mean Time to Failure (MTTF). We also translate the TEAMS model's D-matrix into a Bayesian network (BN), which allows probabilistic diagnostic reasoning and the incorporation of priors on component reliability and failure likelihood.

Proper V&V requires the analysis of the health model to a certain degree of coverage and not just on a few selected and hand-crafted failure scenarios. While our tool set allows for manual specification of fault scenarios, it uses advanced algorithms to automatically generate single and multi-fault scenarios across the entire model or for a selected subset of faults. These scenarios are applied in the context of the mode-sequence commands prescribed in the operational test scripts for the plant. Our tool set uses the mode-enriched fault scenarios to generate the test/mode events from two independent streams—the discrete TFPG model (generated from TEAMS models) as well as a gold standard obtained from a Simulink plant simulation or from Cryo lab experimental data. Comparison of the data generated from the two independent streams allows for cross-validation of the discrete TFPG (TEAMS) model and the high-fidelity physics based Simulink model.

The V&V process is made more rigorous by perturbing a number of independent parameters including time of fault injection, fault magnitude, discretization, and thresholding parameters, among others. Parametric Model Analysis (PMA) provides a rich data set for a detailed analysis of the fault-effect coverage on the tests associated with the fault including analysis of the wrapper code.

This paper demonstrates the tool and its capability on a case study of a NASA cryogenic fuel loading facility.

This paper is structured as follows: after discussing related work, we will present our tool architecture (Section 3). In Section 4, will give a brief overview of the NASA cryogenic fuel loading facility and present a selected subsystem as our example. We then demonstrate sensitivity/robustness analysis, test/model coverage, and the analysis of cross-validation results. Section 5 concludes and discusses future work.

2. RELATED WORK

It is obvious that a fault detection and diagnosis system is a highly safety-critical piece of software. Thus, it needs to undergo rigorous V&V and certification. For example, DO-178C, Sec 2.4.3 (RTCA, 2011) requires that a monitoring device has to undergo V&V to the same level as the system it monitors. Due to its specific structure and the use of non-standard reasoning algorithms, however, traditional V&V techniques are not directly applicable, and only a few approaches toward V&V of fault detection and diagnosis systems have been reported. For example, Lindsey and Pecheur (2004) describe a model-checking approach for Livingston health models that can fully exercise the state space. Schwabacher, Feather, and Markosian (2008) discuss various approaches for the V&V of an advanced FDDR system for a NASA space system; Reed, Schumann, and Mengshoel (2011) describe an approach on systematic analysis (parametric analysis) of a Bayesian FDDR model for ADAPT.

As pointed out in (Schumann, Srivastava, & Mengshoel, 2010; Srivastava & Schumann, 2013), any diagnosis system must be analyzed and validated on both the *model level* and the *implementation level*. Most approaches in the literature aim at model validation; actual testing of the system implementation for code coverage (e.g., MC/DC (RTCA, 2011)) has not been reported yet and is difficult due to the usually table-driven algorithms in this domain. The approach described in this paper addresses both model-level and the implementation level validation—especially for key parameters of the wrapper code and the reasoner engine through cross-comparison with other reasoners.

3. TOOL ARCHITECTURE

Validation of the Systems Health Management (SHM) in safety critical systems through rigorous testing of the deployed diagnosis engines (reasoners) is extremely important for safety and mission success. This process should help to understand the quality and limitations of the current SHM setup and provide relevant guidance to further fine-tune and improve the performance of the health management system.

With this in mind, we have designed our tool suite (Figure 1) that uses the concepts of cross-validation to compare the results of the deployed baseline reasoner against other candidate reasoners that can employ richer models over a multitude of auto-generated test-cases (automatic fault scenario generation), taking into account the realistic variation of key

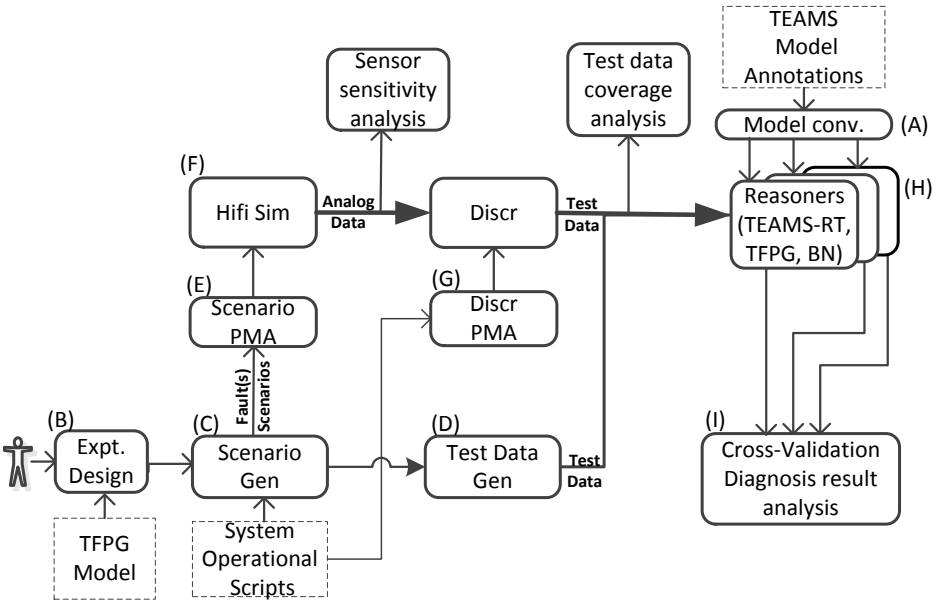


Figure 1. Tool architecture

parameters (parametric model analysis) related to the system (plant, signal preprocessing and discretization). Figure 1 captures the processes and flows across our V&V tool suite. The tool suite uses multiple model-based reasoners such as TEAMS/RT, TFPG, and Bayesian networks.

3.1. Overview

In an initial step (A), the given model, here developed with TEAMS Designer, is translated and prepared for each specific reasoner. While some of these models are basic, others are much richer and can take into account additional details and knowledge available on fault-propagation such as sequencing, timing and mode constraints, or probabilistic information. These models are generated through an automatic translation and annotation process.

The next step (B) is to design the experiment wherein our tool set allows the engineer to specify the required coverage of the test-cases in terms of the complete model or a subset of faults, including single and/or multi-fault combinations. Furthermore, the designer can specify plant operational sequences (commanded mode changes) in which these fault-scenarios need to be tested. Based on the experimental design, the fault-scenarios (C) and their associated ideal test-data (D) are auto-generated using the discretized fault-model.

Alternately, a high-fidelity simulator (F) with fault-injection capabilities is used to generate analog sensor values for each fault-scenario (generated in C). The analog data is then discretized to generate test-data. This process is further enriched by using PMA techniques to generates rich, yet small set of test-cases by perturbing fault magnitude and timing parameters (E), as well as monitoring and discretization parameters (G).

The auto-generated test-cases are then fed to each of the reasoners (H). Their outputs form the basis for the cross validation analysis (I) to get a handle on the diagnosis quality and fault-coverage taking into account the results of the sensor sensitivity analysis and test data coverage analysis. The tool suite is augmented with report generators and a number of advanced statistical analysis and visualization capabilities.

3.2. Reasoning Engines

3.2.1. TEAMS Emulator

Diagnostic reasoning with the given TEAMS model is performed using an implementation of the D-matrix diagnosis algorithm. Given a vector of discrete test results (pass, fail, unknown) and the D-matrix, four sets of failure modes are calculated, those, which are “good”, “bad”, “suspect”, or “unknown”. Failure modes in the suspect list are those, for which some tests have failed, but there has been not enough information for disambiguation.

3.2.2. TFPG

A TFPG model is a labeled directed graph where the nodes represent either failure modes, which are fault causes, or discrepancies, which are off-nominal conditions that are the effects of failure modes. Edges between nodes in the graph capture propagation of failure effects over time in the dynamic system. The model is used for fault diagnostics by collecting observations about anomalies and discrepancies (i.e., tests) in the system, and then using efficient graph search algorithms to generate fault source candidates, i.e., failure modes of components.

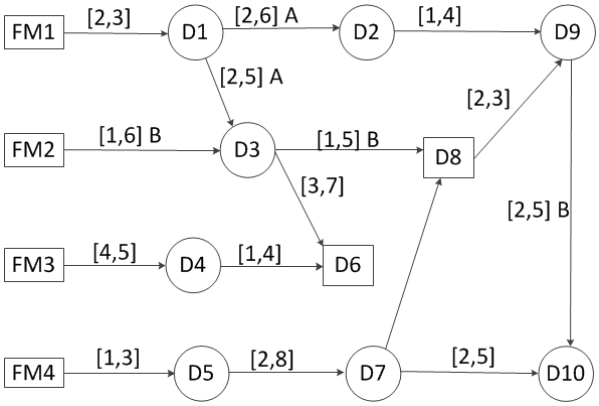


Figure 2. Example TFPG model

Figure 2 shows, as an example, a generic TFPG model. Here, rectangles represent the failure modes (FM1, FM2, ...) while circles represent OR discrepancies and squares AND discrepancies. Edges between nodes capture failure propagation in the system. The edge labels of the form: $[min, max]mode$ capture the failure propagation constraints in terms of timing interval (minimal and maximal expected times) and operational mode(s).

The TFPG modeling approach lends itself to creating system-level, hierarchical fault-propagation models of complex (physical) systems, where component failure modes are anticipated, their failure effects (discrepancies) are observable, a clear cause-effect relationship exists between failure modes and discrepancies, and the failure effects cascade across components (via material, energy, and information flows).

The TFPG reasoner (Abdelwahed, Karsai, & Biswas, 2005; Abdelwahed, Karsai, Mahadevan, & Ofsthun, 2009) employs a robust consistency based diagnosis algorithm that can account for multiple simultaneous faults while taking into account failure propagation constraints based on timing, operational mode(s), and test/effect cascading sequences. The reasoning algorithm is robust to realistic monitoring problems associated with the Tests/Alarms - false-positives, false-negatives and intermittence. The TFPG approach has been applied to and evaluated for various aerospace and industrial systems (Mahadevan & Karsai, 2000–2014; Abdelwahed et al., 2009; Hayden et al., 2006) and recently applied in the context of component-based software system (Abdelwahed, Dubey, Karsai, & Mahadevan, 2011).

3.2.3. Bayesian Networks for HM

Bayesian networks (BN) can be used for diagnosis and decision making. Domain knowledge and probabilistic information about sensor and component reliability, like MTTF, as well as failure likelihood can be easily expressed as priors. We developed a transformation of the given TEAMS model (i.e., the D-matrix) into a Bayesian network, which is inspired by (Pearl, 1988; Luo, Tu, Pattipati, Qiao, & Chigusa, 2005). Optimizations like divorcing and a subsequent translation into arithmetic circuits result in an efficient statistical reasoning engine for large models.

3.2.4. Other Reasoners

Our tool architecture allows us to incorporate additional reasoners, like, for example, HyDE (Narasimhan & Brownston, 2007), which uses simulation over simplified physical models to support diagnostic reasoning. Similarly, systems, like KATE (Goodrich, Narasimhan, Daigle, Hatfield, & Johnson, 2007), which is a generic shell for model-based simulation, monitoring and reasoning, could be added to the set of reasoners for cross validation. In these cases, however, the given TEAMS model cannot be directly translated into a model for those reasoners, as the semantic difference is too large.

3.3. Automated Scenario Generation

The Diagnostic Verification (DVER) tool for the automated scenario generation allows the user to specify the experiment design parameters relative to the appropriate discrete TFPG fault model. The user can specify the set of faults that need to be covered as part of the experiment. The coverage could include the entire model or a specific set of faults in the model. Additional parameters that can be input include: number of faults to be generated per fault scenario (e.g., single-fault, two-fault, etc.), mode change sequence to be applied, timing consideration for the fault propagation interval (e.g., minimal, random, or maximal delay), and number of missing (false-negatives), inconsistent (false-positives) or intermittent tests. Figure 3(left/center) shows a screen-shot of the DVER interface to configure the experiment.

Brute force fault scenario generation involves generating all combinations of faults from the selected list to produce single- and/or multi-fault scenarios. The n-factor algorithm used in Parametric Model Analysis (see Section 3.5) could be used to generate the minimal combinations of fault-scenarios to get the desired fault-coverage. Each generated fault-scenario includes the list of faults and their respective fault-injection times. Test-vector generation for each fault-scenario involves using the TFPG model to simulate the graph traversal starting from the fault-nodes listed in the scenario. The traversal takes into consideration any timing/mode constraint imposed by the TFPG along the fault-propagation sequence. Depending on the user selected option, it chooses the minimal/maximal or a random intermediate time (between minimal and maximal delay) for each propagation link. As the graph is traversed, the triggering time for each node is recorded. The simulator advances the clock to the next time-stamp. The nodes that are marked to be triggered at the time-stamp are then marked visited, and the traversal proceeds to mark the trig-

gering time for its child nodes. When the node corresponding to an observable discrepancy is visited, the triggering time for the test/monitor is recorded. A test once triggered is considered to be latched in that state. Any updates to the graph are applied based on the mode-changes at the specified time. The simulation/traversal process is completed when all possible discrepancy nodes are reached subject to the fault triggering time, propagation time, the mode-change sequence. The test-scenario captures the triggering time for the visited/triggered tests as well as any mode-changes. Missing tests are generated by randomly removing one or more triggered tests from the test-scenario. Inconsistent tests are generated from the set of tests that are not visited during the traversal. Intermittent tests are generated by repeatedly toggling the test-status at random times (within a specified interval).

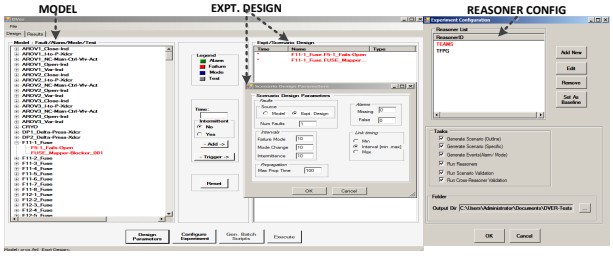


Figure 3. DVER experiment configuration

3.4. Cross Validation

Cross validation of the deployed baseline reasoner results with the results of other candidate reasoning engines facilitates analysis of the correctness, reliability, and limitations of the deployed SHM model and process. As any diagnostic model represents a simplified and abstracted model of the underlying physical plant, we leverage off an abstraction hierarchy, which simplifies the plant model towards different domains. In the current instantiation of the tool suite, the reasoners considered include TEAMS/RT (baseline deployed reasoner), TFPG, and a BN diagnoser. While the TEAMS/RT engine uses a simple dependency matrix between fault and tests in each operating mode, the TFPG and BN reasoners can take into account additional details pertaining to timing, fault propagation (sequence), and probabilistic information, respectively. In the abstraction hierarchy this would mean a step towards the time domain, and the probabilistic domain. The use of HyDE (Narasimhan & Brownston, 2007), which uses simplified physical models to support diagnostic reasoning, would correspond to yet another step in the abstraction hierarchy.

The cross-validation process starts with *Scenario Validation* - validating reasoner results against the ground truth fault-scenario to group the listed faults per hypothesis as well as across all hypotheses to identify the fault sets *Match* (true-positives – match with fault scenario) and *Extra* (false-positives – do not match with fault scenario). These are used to compute metrics that reflect the diagnosis quality in terms of *Degree of Match* (ratio of number of matched faults to total number of scenario-faults) and *Accuracy*. In cross validation, the *Match* and *Extra* sets (computed during scenario validation) of the baseline reasoner is compared against those of a candidate reasoner to compute coverage/confidence metrics that indicate the relative closeness of the correctness (match with ground truth) and accuracy (match in terms of ambiguous or erroneous results) of the two reasoners. The cross validation process is repeated against multiple reasoners to get a better assessment of the relative quality of the baseline reasoner. These results from scenario and cross validation are averaged over the desired/expected scenarios (fault subset, single/multi fault, varying fault magnitude, varying test thresholds) to get an overall assessment of the baseline diagnosis quality. Figure 3(right) shows the screen-shot of the interface for configuring reasoners.

3.5. Parametric Model Analysis

Results of system runs with parametric variations are important, among others, for robustness and sensitivity analysis. Traditionally, methods of single-parameter variation or statistical Monte Carlo techniques are used. These methods, however, fail to work on multi-failure analysis or require a large number of test cases without providing any guarantee for coverage of the parameter space. Our GUI-based PMA tool (Reed et al., 2011; Schumann, Bajwa, Berg, & Thirumalainambi, 2010; Schumann, Gundy-Burlet, Pasareanu, Menzies, & Barrett, 2009) uses an n-factor algorithm for generating perturbed fault scenarios and to modify discretization and timing parameters. For the generation of test vectors, the given perturbation range for each variable is discretized into a (small) number of bins, in our case 5, which could correspond to “almost nominal”, “lower”, “higher”, “much lower”, and “much higher”. Then the n-factor generation picks individual bins for each variable in such a way that (a) each bin of each variable is present at least once, (b) for all pairs of variables, all combinations of their pairs of bins are present. If $n = 3$, condition (b) must hold for all triples. This means that for a given n all m-ary combinations for $m \leq n$ must be present in the test set, but not necessary combinations for larger m. An n-factor algorithm makes the assumption that failures in a

Table 1. N-factor performance for different number of variables. Number of test cases and generation time (in parentheses) shown for calculations under 10 minutes.

variables	n=2	n=3	n=4	n=5
5	35(1s)	180(1s)	775(1s)	3125(1s)
10	45(1s)	309(1s)	1878(9s)	10364(480s)
15	53(1s)	390(1s)	2546(100s)	
20	58(1s)	446(1s)	3046(537s)	
50	74(1s)	629(49s)		
100	85(1s)	784(724s)		

system are only caused by $m \leq n$ triggers, and higher-order combinations of $n + 1$ or greater factors are not necessary. Experience indicates that 2 or 3 factors are usually sufficient for most applications with a substantially reduced number of test cases. Table 1 shows number of generated vectors and the generation times on a Macbook Pro.

Similar effects can be observed when using n-factor for code coverage testing. Giannakopoulou et al. (2011) report that a 3-factor set reduced the size of the test set by more than 3 orders of magnitude compared to the combinatorial exploration. Yet, only about 2% of code coverage was lost. Random test sets of the same size led to a substantially reduced coverage.

3.6. Analysis and Report Generation

Our tool suite can perform a number of analyses regarding sensor and discretization sensitivity and robustness, test data coverage, and cross validation. Since regular health models contain a large number of signals, tests, and failure modes, visualization of results is a challenge. The tool's visualization and analysis capabilities focus on three main areas: sensor sensitivity, test data coverage, and cross-validation. For our tool, we provide several levels of detail, ranging from navigable HTML documents showing the individual time series data for each sensor in a very detailed way to ROC (Receiver Operation Characteristic) curves, which summarize the overall system performance over multiple scenarios in a single plot. The user can interpret the results with a visual interface and assess the quality of the health model to the desired level of detail. We will present results of some of these analyses in the next section.

4. APPLICATION

4.1. NASA Cryo Fuel Loading

Most liquid fuel rockets use cryogenic liquid oxygen LO_2 as oxidizer, which provides high thrust per volume but is difficult to handle. Depending on the size of the rocket, extremely large large amounts of LO_2 must be pumped from a storage tank into the tank of the rocket. The different modes of operation include chill-down phases, filling (slow and fast), as well as draining the pipes, or pumping the LO_2 back into the storage tank in case the launch has been scrubbed.

Figure 4 shows a schematic overview of such a plant; the storage tank on the left-hand side contains the oxygen, from where it is pumped—using several pumps—into the rocket tank. Electrically and pneumatically operated valves control the flow through the various pipes. An operator console is used to control the loading operations and to display results of the health management system. Numerous pressure sensors, temperature sensors, and flow sensors provide real-time information about the plant status.

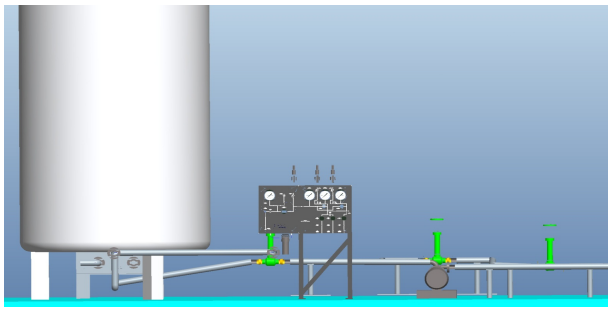


Figure 4. Generic Cryo Fuel loading plant (schematic)

4.2. Health Management and TEAMS Modeling

For this plant, a health management and diagnosis system (Goodrich et al., 2007) is being developed, using the commercial QSI TEAMS modeler and TEAMS/RT diagnosis engine. The plant is instrumented with multiple sensors for pressure, temperature, and flow. These sensor readings are captured at fixed time intervals and preprocessed in the TEAMS wrapper (Figure 5), where the signals are discretized to form test results, which are in turn used by the diagnostic engine. A single sensor can produce several test results, e.g., for a pressure sensor p, there are tests: p-nominal-in-range, p-too-high, p-too-low, etc. The outcome of each test can be "pass", "fail", or "unknown". The TEAMS model, which is based on a hierarchical multi-signal diagnosability analysis consists of several hundred tests and almost 2,000 failure modes, produced diagnosis results as sets of components (or failure modes) known to be "good", "bad", "suspect", or "unknown".

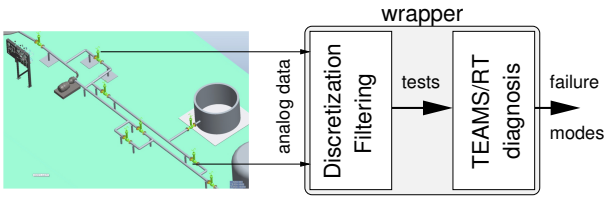


Figure 5. Cryo loading plant with TEAMS/RT wrapper

4.3. Example

For our case study, we consider a small part of a generic cryogenic fuel loading plant (Figure 6). Liquid oxygen is fed from the storage tank (left side, not shown) through the pump. The flow of LO_2 is reduced after the pump by valve V_0. Then a longer pipe transports the LO_2 to the other parts of the plant (right side of the figure). The individual pipes can be drained by means of opening V_1, V_2, or V_3. If V_4 is open, the pipes' LO_2 contents flow into a dump tank, where the liquid oxygen evaporates. This part of the plant is equipped with various pressure sensors p_1, p_2 (red) and a flow sensor f_1 (green/red).

In our operational scenario, LO_2 is pumped and V_0 is partially open to let through the fuel. A constant pressure and flow

can be measured by all sensors. At a certain time t_f, we inject a failure into the system: one of the valves V_1, V_2, or V_3 gets stuck partially open. This fault obviously causes a loss of pressure, because now a majority of the LO_2 is flowing into the dump tank.

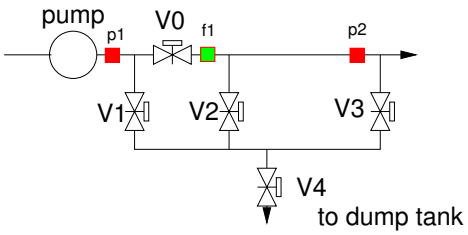


Figure 6. Schematics of small portion of the loading plant with one pump, valves V_1, V_2, V_3, V_4, pressure sensors p_1, p_2 (red), and flow sensor f_1 (green/red)

Figure 7A shows the sensor signals for the two pressure sensors p_1 (left) and p_2 (right), and the flow sensor f_1 (middle). The curves were obtained by running a physics-level Simulink simulator (see Figure 1(F)). Shown are 5 parametric variations of the failure magnitude: the valve gets stuck at $80\% \pm 20\%$. The purple lines are contrasted with a green dashed line showing the nominal (no-fault) condition. The rows of Figure 7A show the scenario, where, V_1, V_2, and V_3 fails, respectively. Graphs of the pressure and flow are shown over time. If V_1 fails, the pressure at p_1 drops almost immediately. The observed pressure drop measured at p_2 is much less and slower, because of the long pipe and the pressure reduction by V_0. The measured flow becomes considerably smaller, because LO_2 back-flows toward V_1. In contrast, when V_2 or V_3 fails, the flow actually increases, because additional LO_2 flows from the pump through the bad valves.

The comparative timing of the signals in these failure scenarios are shown in Figure 7B. The top row shows pressure development over time at location p_1, the bottom row at location p_2, respectively. The settling time ($t_{95\%}$) of the curves, belonging to each scenario can be clearly distinguished. This temporal behavior is caused by physical effects only. For a realistic plant with actual sensors, additional delay times, e.g., caused by W-LAN signal transmission, must be considered.

Our case study will focus on the analysis of these scenarios and the diagnosability of each of the failures. Specific small TEAMS models are used to discuss the tool capabilities.

4.4. Scenario Robustness and Sensitivity Analysis

Parametric Model Analysis on scenarios, shown in Figure 1(E) produces rich data sets that can be used to analyze robustness and sensitivity of the physical plant with respect to the sensors. Only if the value of a sensor changes over time in a characteristic manner when a failure occurs, its output can be potentially used for fault detection.

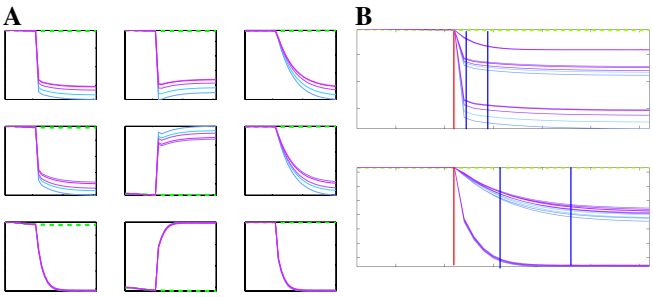


Figure 7. **A**: pressures and flow over time for scenario V_1 (top), V_2, and V_3 (bottom). Left panels show pressure at p_1, middle panels flow at f_1, and pressure at p_2 (right). **B**: delay times $t_{95\%}$ (blue) for pressures at p_1 (top) and p_2 (bottom). Fault injection at t_f shown in red. All results obtained with the Simulink plant simulator.

For a high level of detail, our tool generates navigable HTML reports, which show tables of all parametric variations of the injected faults and the time-series of all sensor outputs, contrasted to a nominal run, similar to the plots shown in Figure 7. For larger systems with many sensors a more compact representation of sensitivity results is needed. For each sensor, we therefore calculate four metrics. S_1: relative maximal deviation of the signal with respect to nominal, S_2: sensitivity of the sensor signal with respect to failure magnitude ($\partial S/\partial F$), S_3: typical shape of the curve (increase/decrease to final value, transient curve, or unspecified), and S_4: settling time $t_{95\%}$. Figure 8A shows the sensitivity for the more than 200 plant sensors for failure scenario V_1. For each sensor, its metrics are shown as star-plots. The length of each side corresponds to the normalized metrics S_1 (red), S_2 (blue), S_3 (magenta), and S_4 (cyan) — see Figure 8(center). Sensors that are not sensitive are shown as light-blue dots. Figure 8(right) displays the differences in sensitivity with respect to scenarios V_1 and V_2. Here, the number of sensitive sensors is much smaller. Sensors, which exhibit a large deviation could be used to disambiguate the failure modes relevant to these scenarios and thus could help to improve the health model.

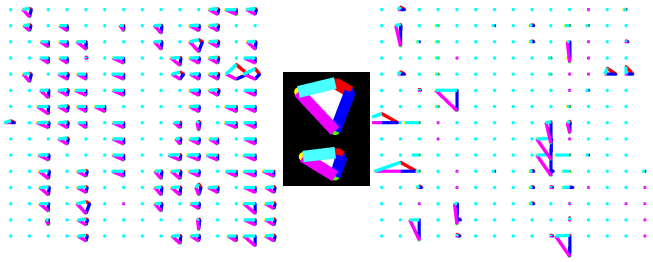


Figure 8. Sensor sensitivity for V_1 fault scenario (left). Enlarged view for 2 sensors (middle). Right panel shows the difference in sensitivity between scenario V_1 and V_2.

4.5. Threshold Robustness and sensitivity analysis

Obviously, discretization thresholds play an important role for the overall performance of the diagnosis system. Therefore, an important task is to analyze if the discretization thresholds that are provided by the domain experts are set appropriately and do not influence the diagnosis result in the presence of noise or variations in the fault magnitudes.

Figure 9 shows plots of two sensor readings S_1, S_2 over time for different failure magnitudes, as obtained from the Simulink simulator via PMA with nominal behavior (dashed green) and sensor values in different hues of blue according to the fault magnitudes. Two failures have been injected at different times t_1, t_2 (vertical purple lines). Sensor S_1 (top panel) is not very sensitive to the failure injected at t_1, but highly sensitive to failure at t_2. It is clearly visible that S_1 is sensitive with respect to the failure magnitude; different PMA runs produce different time series.

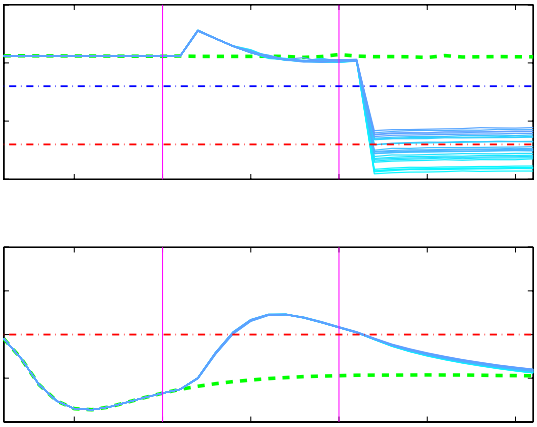


Figure 9. PMA analysis of two sensor signals S_1 (top) and S_2 (bottom)

A threshold, set to the value shown as a red dot-dashed line would result in a situation, where, depending on the actual fault magnitude (which might be subject to noise or other variations), a reliable detection might fail. On the other hand, if the threshold is set to the blue line, the off-nominal situation caused by the failure at t_2 is detected reliable regardless of the fault magnitude.

The bottom panel of Figure 9 shows the output of sensor S_2. Although it is sensitive to failure at t_1, where the nominal and off-nominal traces deviate considerably, no threshold can be found to help to detect this fault. A typical threshold (red line) would flag the fault t_1, but would also trigger during nominal operations (left part of the bottom panel). Note, that this failure causes a transient-style trace, where the value of the sensor goes back to the nominal value after some time despite the fact that this fault has been occurring. The analysis of the proper interaction between sensor signals, discretization, and reasoning results can be performed by the methods described below.

4.6. Test/Model coverage analysis

The test coverage analysis deals with understanding the quality of coverage for each test. This is done by comparing the expected test status against the realistic test status for every fault scenario. The expected test status is based on the failure-effect propagation (reachability) with the discrete fault model that is used by our reasoners (here TEAMS and TFPG). The realistic test status is obtained by thresholding the analog sensor values (from experiment or high-fidelity simulator) for the concerned fault scenarios (possibly across an interesting spectrum of fault magnitude values). The real test status generated for different thresholding criteria is compared against the expected test status to measure the test coverage quality in terms of sensitivity (true positive rate) and specificity (1-false positive rate). A higher test-coverage quality is reflected in terms of high true-positive rate and low false positive rate. The coverage quality for each thresholding criteria may be plotted and compared in an ROC (Receiver Operations Characteristics) curve. Figure 10 below shows the ROC curve obtained by changing the cut-off threshold for the test associated with pressure p_1. ROC curves are typically used to visualize

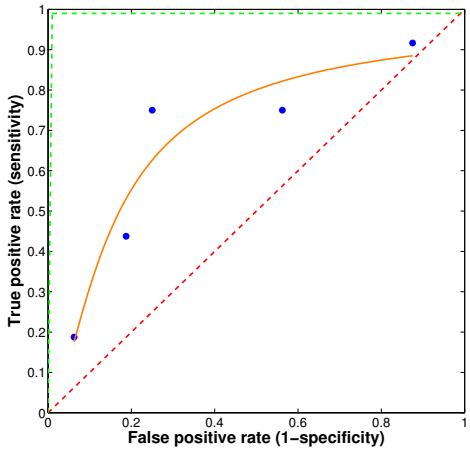


Figure 10. Coverage quality in terms of ROCs. Data points are fitted to $y = 1 - 1/((1 + (x/C)^B)^E$.

the behavior of a clustering or diagnosis algorithm. Results of experiments are shown as points of the true positive rate over the false positive rate. An ideal diagnosis system would be depicted by the green dashed line: a full true positive rate (100%) can be already reached with 0% false negatives. On the other hand, a purely random diagnosis shows up as the diagonal red line.

It is worth mentioning that the above analysis can also help capture any differences in fault-propagation (and thereby triggering of tests) between the discrete fault model (used by the reasoners) and the plant or the high-fidelity simulator. This

is especially true with tests that are never triggered in the realistic scenario but are expected by the fault-model (false negatives), as well as tests that are always triggered but not expected by the fault model (false positives). This shows up in the ROC curve as a shift in the curve along true positive rate axis and/or the false positive rate axis.

4.7. Analysis of Cross-Validation results

Analysis of the scenario validation and cross validation metrics over the specified fault scenarios helps to get a handle on the quality of diagnosis. The scenario validation process compares the faults listed in the scenario ("ground truth") against the faults reported by the diagnoser. The comparison helps identify the true positives (faults that match in the scenario and diagnosis), true negatives (scenario-faults that are not reported by the diagnoser), and false positives (faults listed by the diagnoser that are not part of the scenario). The quality of the results is expressed in terms of *Match* (percentage of true positives among the faults listed in scenario) and *Accuracy* (percentage of true positives among all faults listed by the reasoner). While *Match* is a measure of the ability of the reasoner to identify the real fault sources, *Accuracy* is a measure of the ambiguities listed by the reasoner.

Table 2 captures the results of scenario validation for our case study. It shows *Match* and *Accuracy* of three reasoners—TEAMS emulator, TFPG using the same fault-propagation model as TEAMS, and TFPG*, a TFPG reasoner using an updated fault-propagation model, which includes fault propagation times and fault propagation sequences based on the results of failure analysis shown in Figure 7. Specifically, the TFPG* model has been updated with (a) fault propagation time and (b) a propagation link between p_2 and p_1 for fault from V_3. Table 2 shows the results for ideal test vectors the TEAMS and TFPG model exhibit similar performance, but the TFPG with the updated model has a far greater accuracy (fewer ambiguities). In case of the realistic test vectors that include missing alarms, false alarms, and intermittents, the TFPG reasoner has a slightly higher accuracy probably related to the way intermittents are handled. The TFPG reasoner identifies intermittence and waits for the tests to stabilize before updating results. The TEAMS emulator, on the other hand, starts afresh with every time-stamp. This could also explain the slight decrease in *Match* (compared to ideal) for the TEAMS emulator, as it does not report any faults when all alarms disappear while exhibiting intermittence.

In computing the cross validation metrics, the candidate reasoner results for ground truth and the baseline reasoner results are compared to identity, for each result, the true positive (faults listed by both reasoners), true negative (faults listed by candidate and not by baseline), and false positive (faults listed by baseline and not by candidate). These help analyze the degree of *Match* between the reasoners in identifying faults (*Match Scenario*) and in eliminating ambiguities (*Match Extra*). These metrics help establish the accuracy of the baseline deployed reasoner relative to the candidate reasoners.

Table 3 captures these metrics for the baseline TEAMS emulator relative to the two candidate TFPG reasoners. The high numbers for the *Match Scenario* reflect the closeness between the baseline and candidate reasoner in identifying the source of the fault. A lower *Match Extra* in case of TFPG with the update model reveals that the candidate reasoner has a tighter ambiguity set than the baseline reasoner.

Table 2. Scenario Validation

Reasoner	Ideal Test-Vectors		Realistic Test-Vector	
	Match	Accuracy	Match	Accuracy
TEAMS Emulator	1	0.66	0.9	0.47
TFPG	1	0.66	1	0.59
TFPG*	1	1	1	0.83

Table 3. Cross Validation (baseline - TEAMS Emulator)

Reasoner	Ideal Test-Vectors		Realistic Test-Vector	
	Match Scenario	Match Extra	Match Scenario	Match Extra
TFPG	1	1	0.93	0.79
TFPG*	1	0.33	0.93	0.45

Analysis with the scenario validation metric is important to understand the relative performance of different reasoners. A consistently poor scenario validation metric across all reasoners could indicate a problem with the fault model (inability to isolate a fault), sensor placement, or tolerance boundaries on test coverage. Alternatively, the metric could indicate the effectiveness of one reasoner in certain cases (fault scenarios/modes/robustness to test coverage changes). On the other hand, cross validation helps benchmark the performance of the baseline relative to each candidate reasoner. This would be useful when the real source of the fault is not known and the candidate reasoners have to be used to predict the performance of the baseline reasoner. Since each candidate reasoner might have their own limitations, it is better to cross-validate against a bank of candidate reasoners. Furthermore, analysis over different subsets of faults helps identify where the baseline reasoner might be lacking when compared to the candidate reasoner. This analysis makes it possible to improve the performance of the baseline by adding suitable tests or pseudo-tests.

5. CONCLUSIONS AND FUTURE WORK

For V&V it is essential to ensure robustness and reliability of a health management systems, even more if it is to be deployed in a safety and mission critical environment. In this paper, we have presented a tool set to support V&V of

TEAMS health management systems by employing the paradigms of *cross validation*, where diagnosis results of the TEAMS model are compared with results of other, more advanced reasoners, *automatic fault scenario generation* to support extensive testing and coverage analysis, and *parametric model analysis* to enrich test sets for robustness and sensitivity analysis. We used, as an example, a subsystem of a large NASA cryogenic fuel loading system to demonstrate tool capabilities and to present initial results. A number of specific coverage metrics have been introduced for assessing model quality and model coverage during pre-deployment V&V.

In this paper we have described scaling properties of core algorithms of our integrated V&V tools. For example, n-factor combinatorial test generation scales well with increasing dimension. To provide users with succinct yet meaningful metrics to assess validation, we provide a summary analysis in terms of scenario validation (match, accuracy), cross-validation (match scenario, match extra), and ROC curves. We expect to present our results of using this tool suite on a large system with a rich set of failure effect propagation. In future work we will extend our V&V tool suite to include advanced machine learning algorithms and further statistical analysis in order to provide deeper analysis and to improve quality and robustness of scenario generation and parameter perturbation.

ACKNOWLEDGMENTS

This work was in part supported by NASA's Software Assurance Research Program (SARP), project "Advanced Tools and Techniques for V&V of IVHM Systems". Peter Berg provided a Matlab implementation for TEAMS reasoning.

REFERENCES

Abdelwahed, S., Dubey, A., Karsai, G., & Mahadevan, N. (2011). Model-based tools and techniques for real-time system and software health management. *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*, 285.

Abdelwahed, S., Karsai, G., & Biswas, G. (2005). A consistency-based robust diagnosis approach for temporal causal systems. In *16th International Workshop on Principles of Diagnosis* (pp. 73–79).

Abdelwahed, S., Karsai, G., Mahadevan, N., & Ofsthun, S. C. (2009). Practical considerations in systems diagnosis using timed failure propagation graph models. *Instrumentation and Measurement, IEEE Transactions on*, *58*(2), 240–247.

Giannakopoulou, D., Bushnell, D., Schumann, J., Erzberger, H., & Here, K. (2011). Formal testing for separation assurance. *Ann. Math. Artif. Intell.*, *63*(1), 5–30.

Goodrich, C., Narasimhan, S., Daigle, M., Hatfield, W., & Johnson, R. (2007). *Applying model-based diagnosis to a rapid propellant loading system.*

Gundy-Burlet, K., Schumann, J., Menzies, T., & Barrett, T. (2008). Parametric Analysis of ANTARES Re-entry Guidance Algorithms using advanced Test Generation and Data Analysis. In *Proc. i-SAIRAS 2008.*

Hayden, S., Oza, N., Mah, R., Mackey, R., Narasimhan, S., Karsai, G., Shirley, M. (2006). *Diagnostic technology evaluation report for on-board crew launch vehicle* (Tech. Rep.). NASA.

Lindsey, A. E., & Pecheur, C. (2004). Simulation-based verification of autonomous controllers via Livingstone Pathfinder. In K. Jensen & A. Podelski (Eds.), *Proceedings TACAS 2004* (Vol. 2988, pp. 357–371). Springer.

Luo, J., Tu, H., Pattipati, K., Qiao, L., & Chigusa, S. (2005). Graphical models for diagnosis knowledge representation and inference. In *Autotestcon. IEEE* (p. 483-489).

Mahadevan, N., & Karsai, G. (2000–2014). *Fact tool suite.* `https://fact.isis.vanderbilt.edu/`.

Narasimhan, S., & Brownston, L. (2007). HyDE – a general framework for stochastic and hybrid model-based diagnosis. In *In Proc. of 18th international workshop on principles of diagnosis (DX '07)* (pp. 162–169).

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference* Morgan Kaufmann: .

Reed, E., Schumann, J., & Mengshoel, O. (2011). Verification and validation of system health management models using parametric testing. *Proc. of Infotech@ Aerospace 2011*.

RTCA. (2011). *Do-178c: Software considerations in airborne systems and equipment certification.* Retrieved from `http://www.rtca.org`

Schumann, J., Bajwa, A., Berg, P., & Thirumalainambi, R. (2010). Parametric testing of launch vehicle FDDR models. In *AIAA Space.*

Schumann, J., Gundy-Burlet, K., Pasareanu, C., Menzies, T., & Barrett, T. (2009). Software V&V support by parametric analysis of large software simulation systems. In *Proc. IEEE Aerospace.* IEEE Press.

Schumann, J., Srivastava, A., & Mengshoel, O. (2010). Who guards the guardians? — toward V&V of health management software. In *RV 2010.* Springer.

Schwabacher, M. A., Feather, M. S., & Markosian, L. Z. (2008). Verification and validation of advanced fault detection, isolation and recovery for a NASA space system. In *Proc. PHM 2008.*

Srivastava, A., & Schumann, J. (2013). Software health management: A necessity for safety critical systems. *Innovations in Systems and SW Eng.*, *9*(4), 219–233.

Towards Accreditation of Diagnostic Models for Improved Performance

Anuradha Kodali[1], and Peter Robinson[2]

[1]*SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035*
anuradha.kodali@nasa.gov

[2]*NASA Ames Research Center, Moffett Field, CA, 94035*
peter.i.robinson@nasa.gov

ABSTRACT

The research community mainly concentrates on developing new and updated diagnostic algorithms to achieve high diagnostic performance which is necessary but not sufficient for the diagnostic models that are embedded in software. The focus of this paper is to understand the requirements for accrediting diagnostic system models to meet high performance and safety criticality in case of both models and embedded system (model + software). For embedded systems, models need to be accredited first to allow a more accurate distinction of whether the model or the code within which the model is embedded is the cause of degraded performance. This is because, neither standards for models and simulations (NASA-STD-7009) nor software engineering requirements (NPR 7150.2A) are sufficient to accredit the models in embedded systems. NASA-STD-7009 assesses the correctness of the physics in models and simulations and NPR 7150.2A lists software engineering requirements for NASA systems. Thus, it is important to understand the accreditation standards in terms of performance requirements of models in embedded systems that can smoothly transit from NASA-STD-7009 to NPR 7150.2A. We will discuss interactive diagnostic modeling evaluator (i-DME) as an accreditation tool that provides the performance requirements or limitations imposed while accrediting embedded systems. This process is done automatically, making accreditation feasible for larger diagnostic systems.

1. INTRODUCTION

The research community over prior years has concentrated on developing new and updated diagnostic algorithms to avoid diagnostics with ineffective reasoning. But, most of the times, the real root cause of this ineffectiveness is attributed to incomplete or inaccurate diagnostic models (Simpson, & Sheppard, 1991). The models are incomplete due to the constraints arising from cost (e.g. test design) and system complexity issues. Importantly, with increasing complexity, detailing and bookkeeping of the system becomes very difficult leading to missed information in diagnostic models (Sheppard, & Simpson, 1993). Secondly, the models can be inaccurate because of the following reasons: 1. lack of technical expertise, 2. misunderstanding the existing expertise (documents), and 3. human errors. While human errors are unpredictable; the others can be resolved by precise planning and better documentation at every step of model development. Especially, the first two reasons are categorized as novice and intermediary levels of human knowledge, respectively; but even experts can make errors.

Traditionally, diagnostic modeling is independent of design and manufacturing (Simpson, & Sheppard, 1991). Diagnostic modelers build their models by studying design documents and technical manuals. Here, the physics model is fixed while building diagnostic models and optimizing it for maximum performance. Hence, in early 1980s, there was a strong drive to include diagnostics as an engineering task during system development. For this purpose, testability analysis is strategized to include adding/modifying tests, repacking components to decrease ambiguity, decreasing false-alarms, and improving the observability of certain faults (Simpson, & Sheppard, 1992). Testability analysis, while included in system development, decreases maintenance cost and time, and also improves efficiency of diagnostic models without disturbing system's operational performance by supporting sensor selection and placement.

But, the testability methodology ignores three salient features. Firstly, determining fault modeling (at what level), and the causal relationship between faults and tests are not included for testability analysis. Secondly, while performing testability, the diagnostic algorithm is not included to assess

Anuradha Kodali et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the diagnostic performance; thus there is no remedy for misdiagnosis that is incurred later. Thirdly, no cost-effective repair procedure for the system/diagnostic model is provided. Thus, the best strategy here is to verify and validate the diagnostic model by analyzing all its characteristics (faults, tests, etc.) and inserting the faults via simulation to assess the diagnosis (Sheppard, & Simpson, 1998). Considering these factors, Interactive Diagnostic Modeling Evaluator (i-DME) (Kodali, Robinson, & Patterson-Hine, 2013) is developed as an automatic computer-user interactive tool that proposes cost-effective repair strategies related to fault modeling, test design, and their relationship. This is performed on the D-matrix (Luo, Tu, Pattipati, Qiao, & Chigusa, 2006), an abstract representation of the diagnostic model with causal fault-test relationship in terms of 0's and 1's. Matrix entry 1 represents that the test detects the corresponding fault, otherwise vice-versa. Note that adding/removing tests needs changes in both system and diagnostic models. For the other repairs pertained only to diagnostic models, they can be performed even after system development. But, this is not advisable because the diagnostic models will be implemented in software before the end of system development and it is not easy modifying the software always. Note that software is required to implement the diagnostic models and it is important to certify both the model and software for the same required output.

Columbia Accident Investigation Board (CAIB, 2003) stresses the accreditation (certification) of embedded systems (model + its implementation software, for e.g. TEAMS Designer, TEAMS-RDS (Qualtech Systems Inc.)) to "develop, validate, and maintain physics-based computer models (models in embedded systems)". This process is different from accrediting the models alone. These models are pre-accredited before certifying the embedded system. Such a distinction is important to find out if the model or the code is the cause for degraded performance. For this purpose, we are working to achieve the NASA accreditation standards for models and simulations (NASA-STD-7009), and software engineering requirements (NPR 7150.2A) to make them suitable for embedded systems. But, unfortunately, neither of these standards independently, or combined can provide the necessary standards for all the model-based embedded systems. Clearly, the requirements from models that should be satisfied by the embedded system, the inputs to the accreditation requirements of the software code which implements the model, and the relationship of the model and the code accreditation results needs strict scrutiny and is the focus of this paper. This process is also helpful to not expect from the code performance beyond the limitations of its embedded model. This process becomes tedious with large-scale diagnostics models. Thus, it is important to automatically generate the accreditation requirements to the embedded system via interactive diagnostic modeling evaluator (i-DME). This tool repairs the diagnostic models for better diagnostic performance and then certifies them. As a result the necessary requirements are derived for the diagnostic model's implementation in embedded systems.

Thus, this paper details the general performance guidelines for diagnostic models and the corresponding accreditation process when implemented in software. In Section 2, we will address the building of diagnostic models and best modeling practices. We will also explain i-DME architecture's potential as a model accreditation tool. This tool automatically provides necessary standards information to accredit models implemented in embedded system, thus makes it easier to accredit larger diagnostic models. The NASA standard for models and simulations, and software engineering requirements and their interconnection are studied in order to perform accreditation for embedded systems in Section 3. We will summarize the findings in Section 4.

2. MODELING OF SYSTEM AND DIAGNOSTIC MODELS

In a natural sequence of development, diagnostic modeling follows the system development in parallel. Later, the diagnostic model is implemented in software (embedded systems). It is important to have best practices at every phase of development for the required performance. In this section, we focus on system and diagnostic model development and the corresponding tool (i-DME) to enable best accreditation practices for better diagnostic performance.

2.1. System Modeling

System modeling is an important engineering task which requires adequate planning and skillful implementation. Here, modeling includes developing a combination of conceptual, mathematical, logical and/or computational models. Firstly, the personnel in charge of modeling starts with the specifications required to satisfy the objectives and the mission. Then, the conceptual designs are translated into detailed developmental plans for the molding of hardware. At this stage, the personnel in charge can change the requirements set before to suit practical compulsions. This may lead to changing the basic principles and to refine the existing methods continuously. After this, there will be extensive testing, both manually and through test development, to shift the development into qualification – once simulated and real time series data is available. There will be two types of tests: development tests to verify the components to consistently and reliably perform; quantification tests to determine if the vehicle is suitable to perform its specified mission. The system is intensively verified and validated by detecting design deficiencies and early development failures arising from the unanticipated communication among components. This process includes verifying for authenticity of operating conditions, e.g.

pressure and temperature and efficiency of each component/subsystem performance. The validation testing strategy focusses to build a system that is effective and economically viable. This means the model can be built in a timely fashion within the budget structure that accomplishes the mission objectives (Swenson, & Grimwood, 1989).

2.2. Diagnostic Modeling

While system development ensures design for performance; it is important to design it for field operations via an optimized diagnostic model (Simpson, & Sheppard, 1993). The diagnostic information is extracted from system models via technical manuals and design documents. This knowledge is then used to specify a simplified form of the diagnostic model; this is used for testability analysis and diagnosis later. Even though the system development phase is well documented; building diagnostic models as a separate task is troublesome. By doing this routine as part of system development; time, cost, and efficiency of diagnosis can be improved simultaneously without hindering the system's operational mechanism. For e.g. designing tests early to decrease ambiguity at individual, sub-system, and system levels reduces maintenance cost (Simpson, & Sheppard, 1992) (Sheppard, & Simpson, 1992). Also, via this process, the personnel are forced to not only think about performance, but also focus to recover it from a failure condition. This paper once again advocates practicing diagnostic modeling within system development; thus analyzing the system for diagnosability and testability from its early stages of development.

2.2.1. Fault Modeling and Test Design

The first important task in fault modeling is to determine the level at which the diagnosis is performed (Simpson, & Sheppard, 1992). It can be done at component, or sub-system, or system level. In general the level to which diagnostics should be performed is the level to which repair actions can be taken (e.g. LRU – line replaceable unit, ORU- orbital replacement unit). The symptoms associated with each fault mode are analyzed during FMECA analysis (Sheppard, & Simpson, 1992). The corresponding impact, in terms of criticality of the fault mode on mission success, safety, system performance, maintainability, and maintenance requirements is also analyzed. Correspondingly, tests are designed to detect these faults. High detection and design costs are always considered during test design. Also, with the fault dictionary (D-matrix); the set of dependencies between the tests and the fault modes are determined via simulations, dataflow analysis, logic flow analysis, and traditional, manual circuit analysis (Sheppard, & Simpson, 1992) (Luo, Tu, Pattipati, Qiao, & Chigusa, 2006).

When diagnostic models are optimized for better performance from early stages of system development; analysis for fault mode definitions and optimized test designing is performed. This includes analyzing the model for ambiguity in fault modes and designing tests to reduce it. Similarly, analysis for excess (excess test provides the same information as a combination of other tests (Simpson, & Sheppard, 1992)) and redundant tests is performed by incorporating only essential tests that are required for diagnostics. Instead of restricting the tests to check for proper system functioning; they are also required to isolate faults in the model. Models are made up of nodes and arcs, and the propagation paths for fault models are complicated for a complex system. So, it is always important to carefully generate the fault-test relationship in D-matrix. With addition of new components during system development, this relationship is bound to change and should be updated accordingly.

2.2.2. Accreditation of Diagnostic Models: i-DME

Diagnostic modeling has matured from a simple data and file sharing to computerized automatic designer tools (e.g. TEAMS (Qualtech Systems Inc.)). This necessitates accreditation of diagnostic models and their real-time software implementation. The aim is to reduce mean time to isolate faults and recover systems with highest efficiency (Simpson, & Sheppard, 1991). But, this may not always be the case because of improper understanding of testability information. Certain measures (e.g. ambiguity, operational fault isolation etc.) are extracted from the model to check for testability and accordingly, the systems are redesigned (at initial stages) or repackaged (Sheppard, & Simpson, 1992). Similarly, we have focused on building new tests for improved performance of the diagnostic model in isolating faults. But, adding tests is not always the sufficient solution because it may cause other issues with the system operation and cost effectiveness. This debugging and remedial process is always tedious and is impossible for human efforts. Thus, in the realm of system engineering, i-DME tool is developed to debug diagnostic models at every step of system development and operation. This tool, with the aid of supervised data (data is labeled with corresponding nominal or faulty state), debugs diagnostic models and proposes repair strategies to D-matrix (abstract representation of diagnostic model) by coordinating with the decision maker (user) (Kodali, Robinson, & Patterson-Hine, 2013).

i-DME is defined as a combined process of computer and user decisive mechanisms where computer provides platform of the diagnostic analysis of the system model with the aid of supervised data and the decision maker performs the role of accepting/declining repair strategies based on the analysis of performance metrics and technical expertise (see Figure 1). Five D-matrix repair strategies are identified arranged in ascending order of cost effectiveness. These strategies range from addressing duplicity in faults and tests, repairing the fault universe to accommodate lower/higher level fault modeling (re-define the level of fault modeling

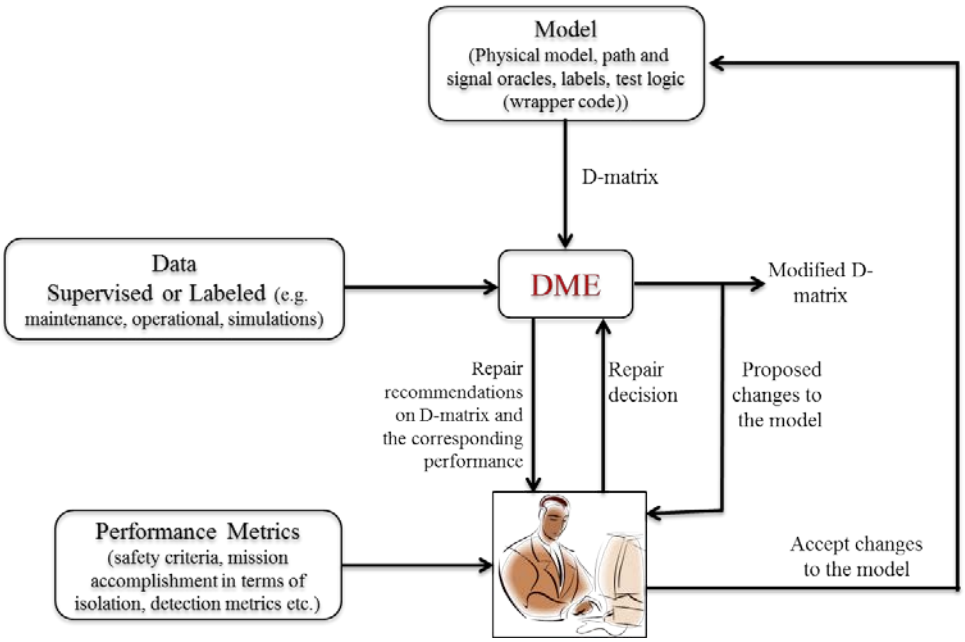


Figure 1. i-DME architecture

by adding or removing rows), repairing/changing the wrapper/test logic, repairing 0's and 1's in the D-matrix entries, and adding/removing tests. They are included in an iterative loop to experiment for better performance along with the decision maker. The performance criteria are based on fault detection and isolation metrics derived from the mission objectives by the user. Then, the decision maker accepts/declines the repair strategies based on before and after performance. More details of this framework can be found in (Kodali, Robinson, & Patterson-Hine, 2013).

In this process, the user not only plays a key role to accept/decline the repair on the diagnostic model, but also prepares the supervised data. The data collected via simulations, maintenance, or operations should be labelled with either nominal or the faulty condition. The credibility of the data depends on skill level of the user. The data can be used to validate the diagnostic model in i-DME process[1]. The system realities which cannot be formalized are also included as user's technical knowledge. Similarly, any diagnostic algorithm which will be employed for diagnosis during operations is implemented in this process for assessing the performance by calculating the corresponding metrics. Importantly, the diagnostic algorithm implemented here for diagnosis is also employed in the software implementation of the system during operations[2].

[1] i-DME efficiency is directly related to the authenticity of the supervised data used for accreditation.

[2] Presently, i-DME is explained for D-matrix; but the framework will be well extended to other modeling paradigms (e.g. fault signature matrix generated using temporal causal graphs (Daigle, Roychoudhury, Biswas, & Koutsoukos, 2010)).

i-DME as an accreditation tool

i-DME not only debugs diagnostic models, but can also double as an accreditation tool for diagnosis and proposes repair strategies to suit the performance. The salient features of this model accreditation tool are listed here:

1. The tool tracks the repairs and diagnostic performance of the diagnostic model throughout the system development and operations, and thus provides important inputs of the performance trends with each repair for higher diagnosability to the modelers (verification and validation).
2. The tool in addition to pointing out the errors or incompleteness in the model provides the strategies about what to do in order to improve the performance.
3. The requirements for system's accreditation are always specified in terms of operation and safety. But, in addition, this tool introduces and derives system requirements in terms of diagnostic performance, viz. detection and isolation metrics when analyzing diagnostic models by including diagnostic reasoning algorithm. It is especially useful to understand the limitations of cost of diagnostic modeling vs performance.
4. The tool adds value by utilizing the advantages of both computer and the decision maker, propose cost-effective repairs that not only include adding/modifying tests, but also corrects the level of fault modeling and causal fault-test relationship; thus investigating all the possible causes of erroneous models.

3. NASA STANDARDS: BRIDGING GAP BETWEEN MODEL AND SOFTWARE ACCREDITATION

In the prior discussion, accreditation process is performed on the models alone by proposing repairs for better diagnostic performance. But, it is also important to certify the embedded systems they are implemented in. This is because, in such a case, it is hard to distinguish if the performance degradation is due to error or incompleteness of the model or software in which it is embedded. For this purpose, test evaluation and execution are evaluated automatically in contrast to the regular practice by hand for software testing (Vaandrager, 2006). The response for each test case is noted when analyzing model against which the embedded system can be tested (Sabetzadeh, Nejati, Briand, & Mills, 2011).

In NASA's context, it is natural to think that the integration of NASA-STD-7009 for models and NPR 7150.2A for software engineering would provide the guidance that is required to accredit embedded diagnostic models. But, to date there is much ambiguity in guidance to accredit embedded model-based systems. In this paper, we focus on accrediting a subset of those systems, viz. diagnostic models.

NASA-STD-7009 provides methods to accredit models, but explicitly states that it does not apply to models and simulations that are embedded in control software, emulation software, and stimulation environments. It also points to NPR 7150.2A, NASA software engineering requirements to apply for such embedded models and simulations. But, in NPR 7150.2A, numerical accuracy, uncertainty analysis, sensitivity analysis, verification and validation for software implementation of models and simulations are stated to be addressed by the center processes and explains that the specific verification and validation information is available in NASA-STD-7009. This is in fact very confusing because NASA-STD-7009 doesn't apply to models and simulations implemented in certain embedded systems. Even for others, as specified in requirements mapping matrix of NPR 7150.2A, models are accredited as per this standard only when they support qualification of flight operations or equipment and ignores for e.g. ground operations/equipment for medium-critical systems (requirement SWE-070 in NPR 7150.2A).

The NASA Software Engineering Handbook (Section 7.15) (NASA software engineering handbook, 2013) recognizes this lack of specific direction and provides additional guidance which states that the analysis of models not covered by NASA-STD-7009 should report requirements 4.2.6, 4.4.1-4.4.9 found in NASA-STD-7009 while implementing NPR 7150.2A. It goes on to state that it is sufficient to merely report on any and all activities performed even reporting that no activities were performed.

For other models, it is important to ensure that the requirements of both the standards (NASA-STD-7009 and NPR 7150.2A) are satisfied. The requirements of NPR 7150.2A are either supplemental, or not related, or subset to the requirements in NASA-STD-7009. In either case, it is important to identify and derive the requirements from the diagnostic models that can be imposed on its embedded implementation. Hence, the process of accrediting embedded diagnostic systems includes 2 tasks: 1) identify the requirements for the accreditation of embedded systems, 2) implement an automated process (i-DME) to derive the requirements (in terms of performance requirements and reports) from the diagnostic model analysis.

3.1. Task 1: Identify the Accreditation Requirements

It is important to identify the input requirements from the model accreditation (NASA-STD-7009) that should be satisfied by the embedded system. This includes documenting the limitations of the model, conceptual details and rationale of the model and test cases, error and warning reports, and credibility scale for the eight assessment factors. The requirement extracted from these documents set the additional new performance requirements for the embedded system. Then, the relationship between the model and the code accreditation results should be scrutinized. This comparison for a similar set of test cases will check if the model is correctly implemented in the software or not. To this effect, we will explain the necessary information derived from model accreditation to the embedded system.

The verification and validation requirements of models as stated in NASA-STD-7009 required for embedded systems are listed as below:

1. Req. 4.4.1 – Shall document any verification techniques used and any domain of verification (e.g., the conditions under which verification was conducted).
2. Req. 4.4.2 – Shall document any numerical error estimates (e.g., numerical approximations, insufficient discretization, insufficient iterative convergence, finite-precision arithmetic) for the results of the computational model.
3. Req. 4.4.3 – Shall document the verification status of (conceptual, mathematical, and computational) models.
4. Req. 4.4.4 – Shall document any techniques used to validate the M&S for its intended use, including the experimental design and analysis, and the domain of validation.
5. Req. 4.4.5 – Shall document any validation metrics, referents, and data sets used for model validation.
6. Req. 4.4.6 – Shall document any studies conducted and results of model validation.

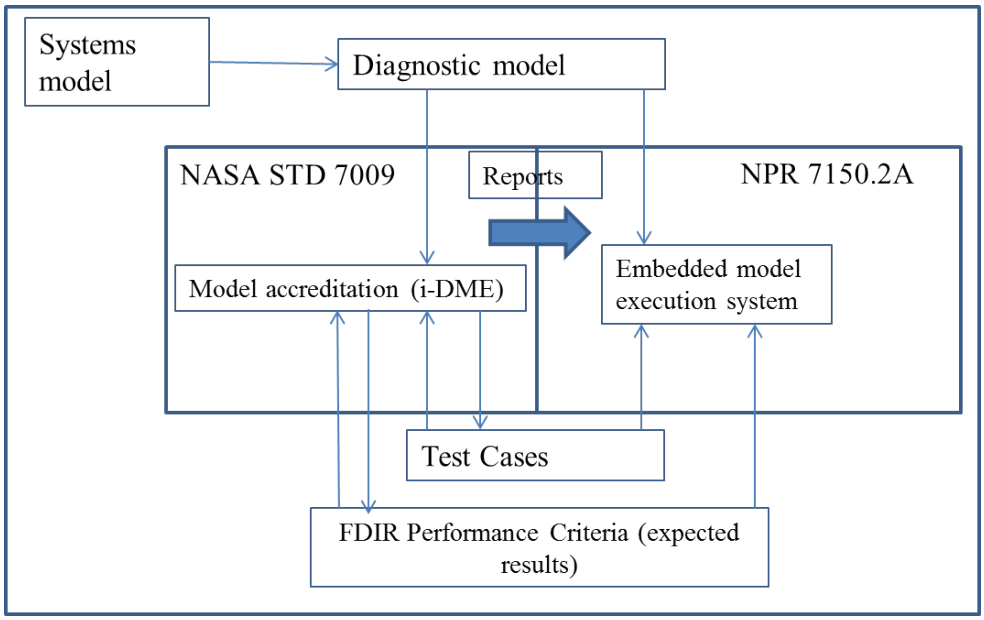


Figure 2. Relationship of i-DME to support 7009/7150.2A integration for embedded diagnostic models

The verification and validation information derived based on these requirements guides the accreditation process of embedded system of how to use and verify the model. Test cases that can be used for accreditation of both model and embedded system are defined and documented (see Figure 2). Similarly, verification and validation techniques (in this case, diagnostic algorithm) need to be the same for both accreditations and should be documented. Using this standard, the diagnostic model is independently accredited and the results are properly documented. In fact, every detail is documented as it is necessary to document everything that is performed or even document that nothing is done.

Analyzing the credibility of the model accreditation process is important to accredit embedded systems. To monitor this, NASA-STD-7009 has a credibility assessment score which is the weighted addition of eight factors, viz. verification, and validation (development), input pedigree, results uncertainty, and results robustness (operations), use history, management, and people qualifications (supporting evidence). These factors scored between 0 and 4 with 4 being the highest score. For e.g. input pedigree gets the highest score when the supervised data mimics the real-world operational data and captures all the necessary problems of interest. Similarly, the decision maker with extensive experience in the use of the diagnostic model corresponds to highest score for people. It is technically feasible, but with difficulty to achieve highest rating and is limited only when the system is in operation, while lower levels can be achieved during early phases of development. The credibility assessment score is documented and reported to the decision maker so that he understands the reliability of the model accreditation results.

Reporting errors and warnings is also a necessary requirement to translate the information from model to embedded system accreditation. During accreditation of diagnostic models; if it is identified that certain repairs to the model cannot be performed due to cost or complexity constraints, then, we document it as a constraint on the performance requirements of the embedded system. Otherwise this deficiency can be attributed to the code while it is being accredited. For e.g. information about components that are not diagnosable with the present model should be documented so that when it is not diagnosed with the working software; wrong manifestation to software can be avoided.

3.2. Task 2: i-DME to Generate Accreditation Requirements

For models of large-scale complex systems, the reporting of the requirements is a huge burden. In addition no specific model assurance activity processes are defined which makes it impossible with laborious manual labor to document the verification and validation requirements. This gap is filled in by the proposed method, i-DME that automatically generates reports for verification and validation requirements in NASA-STD-7009 as stated above. In addition, most importantly, i-DME defines the performance requirements that need to be and can be satisfied by the embedded system derived from the diagnostic model analysis.

The reports for these requirements will be accomplished by running i-DME system on a set of test cases which cover the potential failure sources in the system. For this purpose, as shown in Figure 2, the inputs for model verification and

validation are supervised test cases and user-set performance requirements. Using these, i-DME verifies and validates the diagnostic model by proposing repairs to add new failure modes/tests, or repair the test logic, or repairs the relationship between failure modes and tests in terms of 0's and 1's. After finishing the repair procedure, i-DME assesses the performance and changes the user-set performance criteria to a more realistic assessment. This acts as performance requirement to embedded systems. Similarly, i-DME in coordination with the user develops new test cases or makes corrections to the existing ones when the corresponding labels of nominal or off-nominal conditions are mistaken. All these requirements, test cases, and performance, are in line with those in NASA-STD-7009 and are documented in a user-friendly manner by the i-DME. The details about the diagnostic algorithm used for performance assessment will also be provided because it is mandatory to use the same technique while accrediting the model and the embedded system.

The capabilities of i-DME in the context of NASA-STD-7009 and NPR 7150.2A for the accreditation of models are listed below:

1. i-DME is an automated performance reporting tool. Thus, it becomes easier to accredit even very large scale diagnostic systems.
2. i-DME provides a framework to benchmark the diagnostic models against supervised data ("test cases"). These same test cases will also run against the code.
3. For verification and validation, the diagnostic algorithm calculates the performance in terms of detection and isolation metrics. This is also used to assess the credibility of the models for accreditation.
4. The system's faulty behavior as assessed by the diagnostic model is reported to the decision maker on a regular basis.
5. The limitations of the diagnostic model, for e.g. cannot achieve 100% isolation with insufficient tests, are obtained via i-DME process through the reporting to the decision maker. This avoids imposing incorrect performance requirements while accrediting embedded systems.

Conclusively, the diagnostic models and simulations are pre-accredited based on NASA-STD-7009 and then accredit the embedded system based on NPR 7150.2A by automatically deriving necessary requirements via i-DME. This enables clear distinction of the reason for performance degradation even in large-scale embedded systems. Also, by doing this, we understand what not to expect from the embedded system beyond the capabilities of the implemented model. This is because these limitations can be manifested as erroneous implementation in the code. Note that, diagnosing for errors in software code is not the focus of this paper.

3.3. Accreditation Requirements for ADAPT System

We demonstrate i-DME framework as an accreditation tool on ADAPT system (Poll, Patterson-Hine, Camisa, Garcia, Hall, Lee, Mengshoel, Neukom, Nishikawa, Ossenfort, Sweet, Yentus, Roychoudhury, Daigle, Biswas & Koutsoukos, 2007). During accreditation of D-matrix using i-DME framework, repairs are proposed to the D-matrix entries corresponding to voltage and current sensors of component FAN (underspeed and overspeed failure modes) to avoid misdiagnosis. This process is already published in (Kodali, Robinson, & Patterson-Hine, 2013) and is not presented here.

The information derived from ADAPT model accreditation needs to be reported for embedded system accreditation. The user sets correct isolation rate as the performance requirement on the model. Correct isolation rate is the percentage number of events that are correctly diagnosed (both nominal and faulty cases) over time. This metric is reported for each failure mode and nominal case whenever supervised data is available (see Figure 3). Note that, the performance requirement is based on user's decision and i-DME analyzes the model based on that metric. The diagnostic algorithm used during model accreditation, DMFD algorithm (Singh et al., 2009) is also reported. i-DME reports the performance requirements for embedded system accreditation as shown in Figure 3. The performance details (correct isolation rate) for each failure mode and nominal conditions against the given test cases along with the repair conditions proposed to achieve the corresponding performance are reported. These metrics are used to set requirements for comparison check for the available test cases when the software implementation of ADAPT diagnostic model is accredited.

4. CONCLUSIONS

In this paper, the accreditation process for diagnostic models and the corresponding embedded systems is discussed. It is important to include building of diagnostic models during system development so that any changes to the system model for better diagnosability can be proposed early. In this perspective, to debug diagnostic models at every step of development and operations, i-DME tool can be employed. As an accreditation tool, i-DME also proposes repairs on the diagnostic/system model that achieve better performance. Importantly, i-DME also pre-accredits the diagnostic model embedded in software systems and derives the corresponding necessary accreditation requirements for the embedded system. This facilitates isolating the root cause if the model or the code within which the model is embedded is the cause of degraded performance in the case of embedded systems. This is necessary as NASA standards, viz. NASA-STD-7009 and NPR 7150.2A, have restrictions to accredit all the embedded systems. For this purpose, process to translate knowledge from model accreditation to

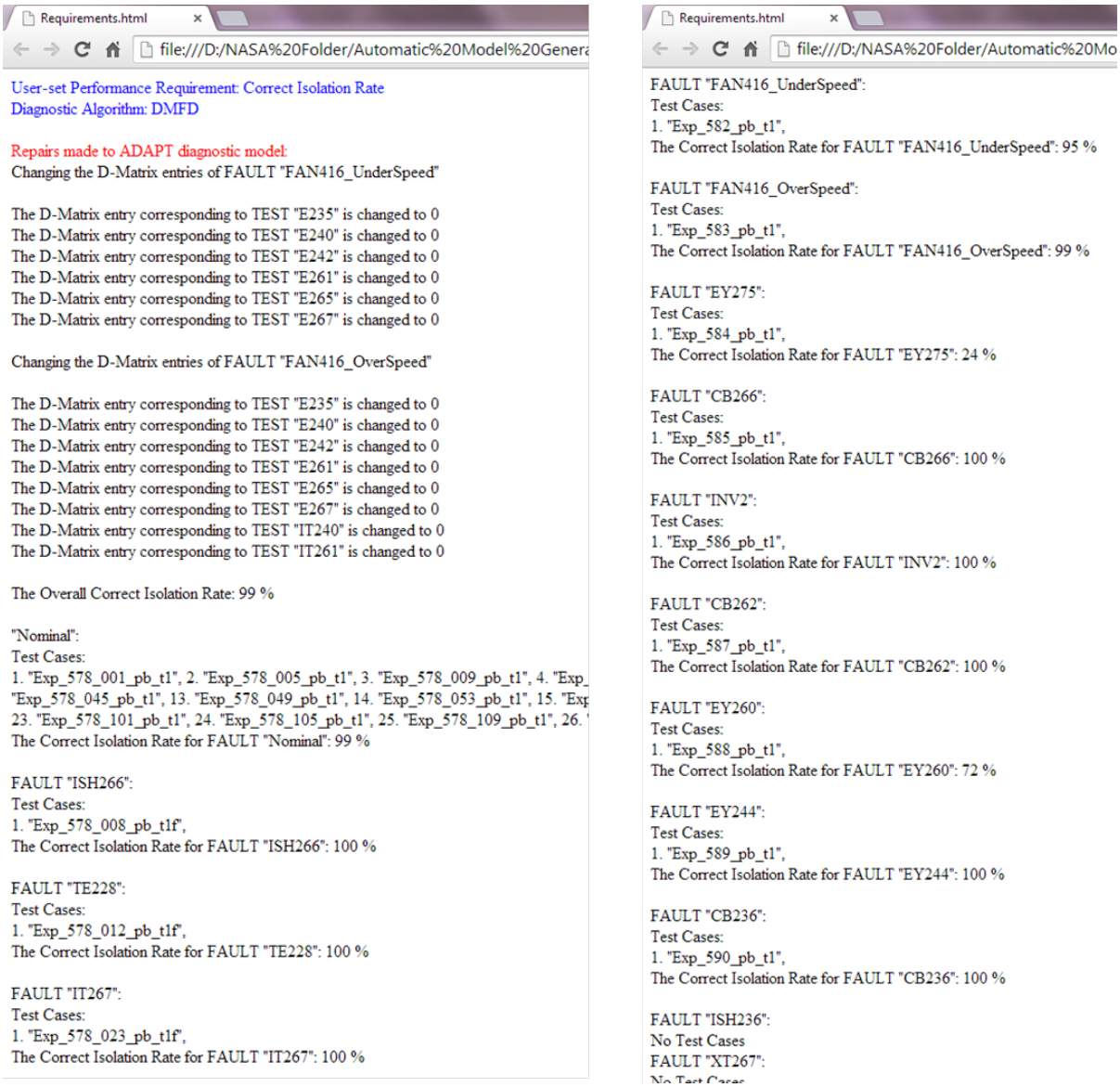


Figure 3. Reporting of accreditation requirements for embedded ADAPT system

embedded system accreditation as requirements is defined. i-DME automatically generates for verification and validation requirements, thus making it possible to accredit even very large-scale embedded diagnostic systems. In the future, we will explore for uncertainty requirements (requirements 4.4.7 – 4.4.9 in NASA-STD-7009) and credibility assessment score that are necessary for accrediting embedded systems and implement them in i-DME.

REFERENCES

CAIB (2003). Columbia Accident Investigation Board Report. . vol. 1.

Daigle, M., Roychoudhury, I., Biswas, G., & Koutsoukos, X (2010). An event-based approach to distributed diagnosis of continuous systems. *Proceedings of the 21st International Workshop on Principles of Diagnosis*, pp. 15-22.

Kodali, A., Robinson, P., & Patterson-Hine, A. (2013). A framework to debug diagnostic matrices. *Annual Conference of the Prognostics and Health Management Society 2013*, October 14 - 17, New Orleans, LO.

Luo, J., Tu, H., Pattipati, K., Qiao, L., & Chigusa, S. (2006). Graphical models for diagnostic knowledge representation and inference. *IEEE Instrum. Meas. Mag.*, vol. 9, no. 4, pp. 45–52.

NASA-STD-7009 (2008). Standards for models and simulations. NASA, https://standards.nasa.gov/documents/viewdoc/3315599/3315599.

NASA software engineering handbook (2013). *NASA Technical Handbook*. NASA, http://swehb.nasa.gov/display/7150/7.15+-+Relationship+Between+NPR+7150.2+and+NASA-STD-7009#_ tabs-1.

NPR 7150.2A (2009). NASA software engineering requirements. NASA, http://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=7150&s=2.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D., Lee, C., Mengshoel, O., Neukom, C., Nishikawa, D., Ossenfort, J., Sweet, A., Yentus, S., Roychoudhury, I., Daigle, M., Biswas, G., & Koutsoukos, X. (2007). Advanced diagnostics and prognostics testbed. *In Proc. DX'07*, pp. 178–185.

Qualtech Systems Inc., www.teamqsi.com.

Sabetzadeh, M., Nejati, S., A., Briand, L., & Mills, A. E. (2011). Using SysML for modeling of safety-critical software–hardware interfaces: Guidelines and industry Experience. *IEEE 13th International Symposium on High-Assurance Systems Engineering*.

Sheppard, J. W., & Simpson, W. R. (1991). A mathematical model for integrated diagnostics. *IEEE Design and Test of Computers*, vol. 8, no. 4, pp. 25 – 38.

Sheppard, J. W., & Simpson, W. R. (1992). Applying testability analysis for integrated diagnostics. *IEEE Design and Test of Computers*, vol. 9, no. 3, pp. 65 – 78.

Sheppard, J. W., & Simpson, W. R. (1993). Performing effective fault isolation in integrated diagnostics. *IEEE Design and Test of Computers*, vol. 10, no. 2, pp. 78 – 90.

Sheppard, J. W., & Simpson, W. R. (1998). Managing conflicts in system diagnostics. *IEEE Computer*, vol. 31, no. 3, pp. 69 – 76.

Simpson, W. R., & Sheppard, J. W. (1991). System complexity and integrated diagnostics. *IEEE Design and Test of Computers*, vol. 8, no. 3, pp. 16 -30.

Simpson, W. R., & Sheppard, J. W. (1992). System testability assessment for integrated diagnostics. *IEEE Design and Test of Computers*, vol. 9, no. 1, pp. 40 -54.

Simpson, W. R., & Sheppard, J. W. (1993). Fault isolation in an integrated diagnostics. *IEEE Design and Test of Computers*, vol. 10, no. 1, pp. 52 -66.

Singh, S., Kodali, A., Choi, K., Pattipati, K., Namburu, S., Chigusa, S., Prokhorov, D.V., & Qiao, L. (2009). Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques. *IEEE Trans. Syst., Man, Cybern*. A, vol. 39, no. 1, pp. 160–176.

Swenson, Jr., L. S., & Grimwood, J. M. (1989). *This new ocean: A history of project Mercury*. Published as NASA Special Publication-4201 in the NASA History Series.

Vaandrager, F. W. (2006). Does it pay-off? model-based verification and validation of embedded systems!. *In F. A. Karelese (editor), PROGRESS White Papers*.

A Model-Based Approach for Reliability Assessment in Component-Based Systems

Saideep Nannapaneni[1], Abhishek Dubey[2], Sherif Abdelwahed[3], Sankaran Mahadevan[4], Sandeep Neema[5]

[1,4]*Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN, 37235, USA*

saideep.nannapaneni@vanderbilt.edu
sankaran.mahadevan@vanderbilt.edu

[2,3,5]*Department of EECS/ISIS, Vanderbilt University, Nashville, TN, 37235, USA*

dabhishe@isis.vanderbilt.edu
sherif@isis.vanderbilt.edu
sandeep@isis.vanderbilt.edu

ABSTRACT

This paper describes a formal framework for reliability assessment of component-based systems with respect to specific missions. A mission comprises of different timed mission stages, with each stage requiring a number of high-level functions. The work presented here describes a modeling language to capture the functional decomposition and missions of a system. The components and their alternatives are mapped to basic functions which are used to implement the system-level functions. Our contribution is the extraction of mission-specific reliability block diagram from these high-level models of component assemblies. This is then used to compute the mission reliability using reliability information of components. This framework can be used for real-time monitoring of system performance where reliability of the mission is computed over time as the mission is in progress. Other quantities of interest such as mission feasibility, function availability can also be computed using this framework. Mission feasibility answers the question whether the mission can be accomplished given the current state of components in the system and function availability provides information if the function is available in the future given the current state of the system. The software used in this framework includes Generic Modeling Environment (GME) and Python. GME is used for modeling the system and Python for reliability computations. The proposed methodology is demonstrated using a radio-controlled (RC) car in carrying out a simple surveillance mission.

Saideep Nannapaneni et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

In recent years, model-based design (Schattkowsky & Muller 2004; Mosterman, 2007), which is a simulation-based approach, has become a powerful framework for the design of complex systems using component behavior models. It is also used to analyze and manage the complexities and failures due to component-to-component interactions during the design phase of the system. Several design alternatives are possible for the same system and a single design is to be chosen based on several factors such as cost, performance, reliability. Each design choice is associated with a different cost, performance, reliability. The selection of a particular design is made through a tradeoff between the cost, performance and safety of the system. (eg., In an inertial measurement unit (IMU) (Dubey, Mahadevan & Karsai 2012) used in Boeing aircraft, 6 accelerometers are provided even though only 4 are necessary to improve the reliability under additional costs). For commercial airplanes where people are involved, safety takes preference over performance and cost. For unmanned vehicles where people are not involved, performance might take preference over safety. Each design alternative is tested under several scenarios before the final design alternative is selected. A scenario is termed as mission in this paper. A mission can be understood as a collection of activities or functions to be performed. A more formal definition of a mission is provided in Section 4.

Usually, mission requirements are independent of the systems used to undertake the mission. The components used to accomplish the mission functions are indigenous to the system that is carrying the mission. As an example, a simple

mission description can be to move from point A to point B. There can be many choices to move from A to B such as using a gas-powered car or an electric car. The components used in the gas-powered car (fuel-tank, engine) are completely different from the components used in the electric car (batteries) to carry out the same function. In general, not all the components in the system are used to carry out the mission. A system may provide many more functions that are not necessary for the mission. In such cases, all the components corresponding to those functions will be unused and do not appear in the reliability assessment. Assume that B can be reached from A without taking any diversion. In such a case, the steering wheel component will be unused and does not appear in reliability assessment.

Reliability assessment in component-based systems provides a mechanism to predict the failure probabilities for the overall system from the failure probabilities of individual components (Kececioglu, 1972; Krishnamurthy & Mathur, 1997). It is used to evaluate design feasibilities, compare design alternatives, identify potential failure areas in design, trade-off between design factors, provide an insight on the need for redundant systems, and replace existing systems with better reliable systems (Elsayed, 2012). There are two types of mechanical components – repairable and irreparable components. Repairable components are the components that if failed can be brought to working condition. Similarly, irreparable components cannot be brought back to the working state when failed. In the case of repairable components, Mean time between failures (MTBF) is a measure of reliability whereas Mean time to failure (MTTF) is a measure of reliability for irreparable components (Wood, 2001). In this paper, all the components are assumed to be irreparable. Reliability assessment is essential before the beginning of mission and also during the mission. Reliability assessment during the mission is necessary to calculate the reliability of the mission in real-time during the mission in the presence of failure of any of the components. This provides an idea on the redundancy available in the system and assists in real-time decision making process.

Some of the traditional techniques used for system reliability assessment include Failure Modes, Effects and Criticality Analysis (FMECA; Bauti & Kadi, 1994; Teng & Ho, 1996), Fault Tree Analysis (FTA; Lee, Grosh, Tillman & Lie, 1985), Event Tree Analysis (ETA; Ericson, 2005), Reliability Block Diagrams (RBD; Elsayed, 2012), Probabilistic Risk Assessment (PRA; Modarres, 2008; Greenfield; 2001). FMECA is an extension to Failure Modes and Effects Analysis (FMEA) developed by NASA to improve the reliability of space hardware program. In this method, all the potential failures in the design are identified and their severity on the system output is included. In FTA, the system is represented in a hierarchical form using Boolean logic such that the system output occurs at the top. For each system failure, the causes are inferred using a top-down approach. Event trees are used to follow a sequence of events from an initiating event of a component until the end state of the system. The probability of the outcome of end state is determined from the probabilities of individual events. In the RBD approach, the system is represented using a network diagram of blocks representing components connected in series and/or in parallel. The PRA approach uses fault tree and event tree diagrams in a probabilistic framework to compute the probability of a failure outcome. In this paper, reliability assessment is performed using reliability block diagrams because they can be constructed easily using the Boolean expressions employed in the proposed methodology. A detailed introduction to reliability block diagrams is provided in Section 2.

The main contribution of this paper is the extraction of the components involved in carrying out the mission and then constructing the mission-specific reliability block diagram to compute the reliability of the mission using the reliability information of the components in the system. Also, a procedure to extend the proposed methodology to real-time reliability assessment is provided.

The paper is organized as follows. Section 2 discusses the reliability modeling of mechanical components and the procedure for construction of the reliability block diagram. Section 3 provides the details of systems for which the proposed methodology can be applied. In Section 4, the proposed methodology for reliability assessment in component-based systems is presented. In Section 5, the proposed methodology is demonstrated using an example in which a radio-controlled (RC) car is used to carry out a simple surveillance mission. Concluding remarks are provided in Section 6. A list of necessary definitions are provided in the appendix.

2. BACKGROUND

2.1 Reliability Modeling of a Component

A typical component is subjected to three kinds of failures during its service life – (1) early life failures, (2) random failures, and (3) wearout failures. The failure rate corresponding to the early-life failures decreases as a function of service time of component. Random failures are characterized by constant failure rates because failures can occur at any time during the service time of the component. Wearout failures are characterized by an increasing failure rate, where the failure rate of a component increases with the service time of the component. The total failure rate at any time instant is equal to the sum of all the three failure rates. The total failure rate can be modeled using a bathtub curve.

Figure 1 shows a typical failure rate curve for a typical component (Filliben, 2002). The bathtub curve consists of three phases. In the first phase, the early-life failures are

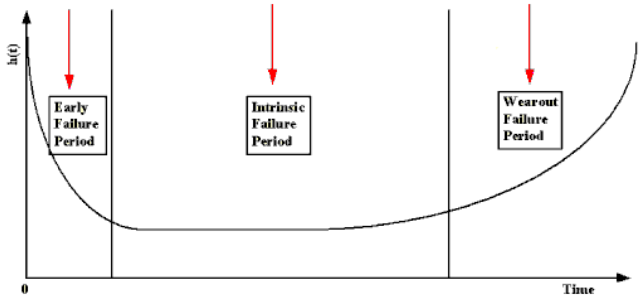


Figure 1. Bathtub curve showing failure rate of a component

predominant; this is known as infant mortality period. In the second phase, random failures are predominant and this phase is known as stable failure period or intrinsic failure period. In the third phase, wearout failures are predominant and this phase is known as wearout failure period. The failure probability during the third phase is generally modeled using a Weibull distribution (Eq. 1) and that during the second phase is modeled using an exponential distribution (Eq. 2). The first phase does not have a failure probability evaluation but early failures are used for design and development.

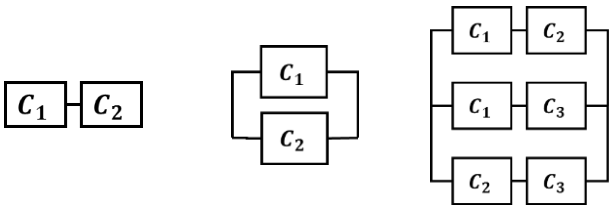
$$P_f(t) = 1 - e^{\left(-\frac{t}{\eta}\right)^{\beta}} \quad (1)$$

$$P_f(t) = 1 - e^{-\lambda t} \quad (2)$$

In Eq. (1), η represents the scale parameter (time at which the failure rate is 0.632) and β represents the shape parameter. The shape parameter describes how the failure rate varies with time. In Eq. (2), λ represents the mean time between failures (MTTF). The values of these parameters can be obtained from the manufacturer, historical data or can be estimated through simulations. In this paper, all the components are assumed to be in the second phase of random failures.

2.2 Reliability Block Diagrams

A reliability block diagram is a graphical representation showing the logical connections between the components in the system. These diagrams are used to compute the overall reliability of the system/functions using the reliability information of individual components and Boolean rules of combinations (Bennetts, 1982). When two components are connected in series, then the function requires both the components and if the components are connected in parallel, either of the components is sufficient to carry out the function. The terms series and parallel carry the same meaning as in the electrical circuits. Figures 2(a) and 2(b) shows series and parallel connections for two components $C_1$and C_2. When components are connected in series, the overall reliability is the product of individual reliabilities of components assuming independence between components (Eq. 3). When components are connected in parallel, the overall reliability is obtained using the union rules from set theory. Also assuming independence between components the expression for overall reliability is obtained using Eq. (4).



(a) Series (b) Parallel (c) r from n

Figure 2. Series and Parallel connections of components

$$R(S) = R(C_1) \times R(C_2) \quad (3)$$

$$R(S) = R(C_1) + R(C_2) - R(C_1)R(C_2) \quad (4)$$

In Eq. (3) and Eq. (4), $R(S)$, $R(C_1)$, $R(C_2)$ refer to the reliabilities of the overall system, components C_1 and C_2 respectively. When the component requirement for a function is specified using "r from n" operator, then all possible combinations are obtained and connected in parallel. The reliability of this component-system is calculated using series and parallel connection rules as stated above. The number of combinations is equal to ${}^{n}_{r}C$, which is equal to $\frac{n!}{(n-r)!r!}$ .Consider an example where a function F requires two out of available three components. Let the three components be C_1, C_2, C_3. In this case, F can be carried out using C_1, C_2 or C_2, C_3 or C_1, C_3. The combinatory can be represented in the reliability block diagram as shown in Figure 2(c).

3. SYSTEM MODEL

The systems under consideration are mechanical systems or cyber-physical systems (CPS). Though, CPS have both mechanical and software components, we currently consider the reliability and failure possibility of mechanical systems only. Software components are assumed to be functional. Consideration of software component reliability metrics require additional future work as these components do not typically age as mechanical components and do not follow the typical bathtub curve. All the mechanical components are assumed to be in the second phase of the bathtub curves, where the failures are random ie the failure rates are constant and the failure probabilities are modeled using exponential distributions. Also, it is assumed that the failures in the components are independent, thus the failure of one component does not influence the functioning of other components in the system. Once a component fails in the system, it remains in the failed state till the end of mission.

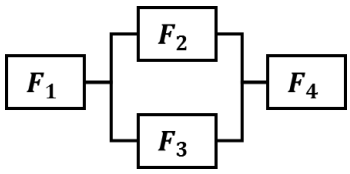
Also, it is assumed that the Mean Time to Failure (MTTF) information is available for all the components in the system.

4. PROPOSED METHODOLOGY

In this section, a step-by-step procedure is developed demonstrating the proposed methodology for reliability assessment.

Step 1. System Modeling: The system undergoing the mission is modeled using a domain-specific modeling language (DSML). The procedure for modeling is not discussed and out of the scope of this paper. The proposed methodology is independent of the language used for modeling. During modeling, each component in the model is associated to the list of functions that require this component. Each component is associated with a corresponding MTTF (mean time to failure) value. The MTTF values for all the components are assumed to be available for analysis.

Step 2. Functional Decomposition: From the mission description, the function-time diagram can be obtained which provides information about the list of high-level functions required and the time when they are required during the mission. (Consider Figure 4. Assume a hypothetical mission description that requires the car to move from A to D. To accomplish the mission, the car which initially is along the line AB should take a left at A, move forward from A to C, take a right turn at C, move forward from C to D. Let the car takes 't_{left}' min to turn and 't_{AC}' min to move from A to C. Therefore, from time t = 0 to t = t_{left}, the high-level function required is to turn left. From t = t_{left} to t = t_{left} + t_{AC}, the high-level function of moving forward is required. Thus, function-time information can be obtained from mission description. This information when represented by a diagram as shown in Figure 6 becomes a function-time diagram). For each of the high-level functions, functional decomposition is carried out to obtain the leaf-level functions. The high-level function can be hierarchically represented in terms of lower level functions and leaf functions using a tree-structure, as shown in Figure 7. From the tree-structure, a Boolean expression for the high-level function can be obtained in terms of the leaf-level functions. This Boolean expression can be converted to a reliability block diagram. The symbol ∧ represents series connection (i.e., both components are needed) and ∨ represents parallel connection (i.e., one of the components is needed). For example, consider a high-level function F which is expressed in terms of leaf-level functions as $F_1 \wedge (F_2 \vee F_3) \wedge F_4$. This Boolean expression when expressed as a reliability block diagram becomes



Step 3. Function-Component association: Each of the leaf-level functions is associated with a component or a component assembly in the system that is undertaking the mission. The components associated with each function depend on the system that is undertaking the mission. The components providing the same function may be different in different systems. (Eg., the power generation function can be accomplished through a battery or an internal combustion engine). A component may be associated with more than one leaf-level function. For each leaf-level function, the corresponding set of components can be derived from GME because in the modeling stage, the association of each component to the list of functions has been made. Again the function-component associations can be expressed using Boolean expressions, which can be extended to obtain the corresponding reliability block diagrams as stated in Step 2.

Step 4. Reliability Assessment: Each leaf-level function has a set of components associated with it and a reliability block diagram can be obtained from the connections of the associated components. Apart from the function-component associations, there are additional constraints called implication constraints (Mahadevan, Dubey, Balasubramanian & Karsai, 2013) that arise from the system model. For example, consider a simple function of power generation in an automobile, which requires an internal combustion engine. When the function-component association is made, the power generation will be associated with the internal combustion engine. But for the working of internal combustion engine, additional components like chassis are required to hold the combustion engine for it to be working. If the chassis breaks down, even though the engine is in working state, the function becomes unavailable. This is an additional implication constraint coming from the system model. Therefore, these implications should also be included in constructing the reliability block diagram. The reliability block diagrams of all the leaf-level functions are used to obtain a reliability block diagram of the high-level function. Similarly, reliability block diagrams can be obtained for all the high-level functions. The reliability block diagrams of all the high-level functions can be combined to obtain the reliability block diagram of the entire mission. Sometimes a component may be required for several function in the mission, therefore the component appears several times in the Boolean expression. The PyEDA package available in Python environment is used here to simplify the Boolean expression and from the simplified Boolean expression, a

simplified reliability block diagram can be obtained. From the mission description, we can obtain the required functions and also the time each function is required for. Using this function-time information, we can calculate the time each of the components is required for. Using the time information, MTTF values and the reliability block diagram, the reliability of the mission can be calculated using series and parallel connection rules given in Eqs. (3) and (4).

Step 5. Real-Time monitoring for decision making: During the course of the mission, the health of all the components can be monitored (failed, or working). If a component is in failed state, all the functions that the component is associated with will not be available. From the health of the components, availability or unavailability of the functions can be inferred. Mission feasibility, as defined in the previous section, can also be analyzed using the health of the components. At any time instant, real-time reliability assessment of the system can be carried out using Step 4. Using the results of real-time reliability assessment, decisions on continuing the mission, aborting the mission or carrying out a simpler mission (a mission with lower outcomes than originally intended) can be made. Also, decisions in choosing alternate paths to maximize the reliability of the mission can be made. When a component becomes unavailable, it can be specified in PyEDA, and it produces a resultant Boolean expression by removing the unavailable component(s). The resultant Boolean expression can be used for reliability assessment of the mission. Figure 3 shows the proposed methodology for reliability assessment.

In Figure 3, the mission is described using high level functions F_1, F_2, F_3, F_4. Then, using functional decomposition, the high level functions are decomposed to leaf-level functions. Then each of the leaf-level functions F_k (k = 5 to 14) is associated to its component assembly. The function-component association also represents the reliability block diagram of the leaf-level function. The reliability block diagrams of the leaf-level functions are combined to obtain the reliability block diagram of the high-level functions. The reliability block diagrams of all the high-level functions are combined to obtain the reliability block diagram of the mission.

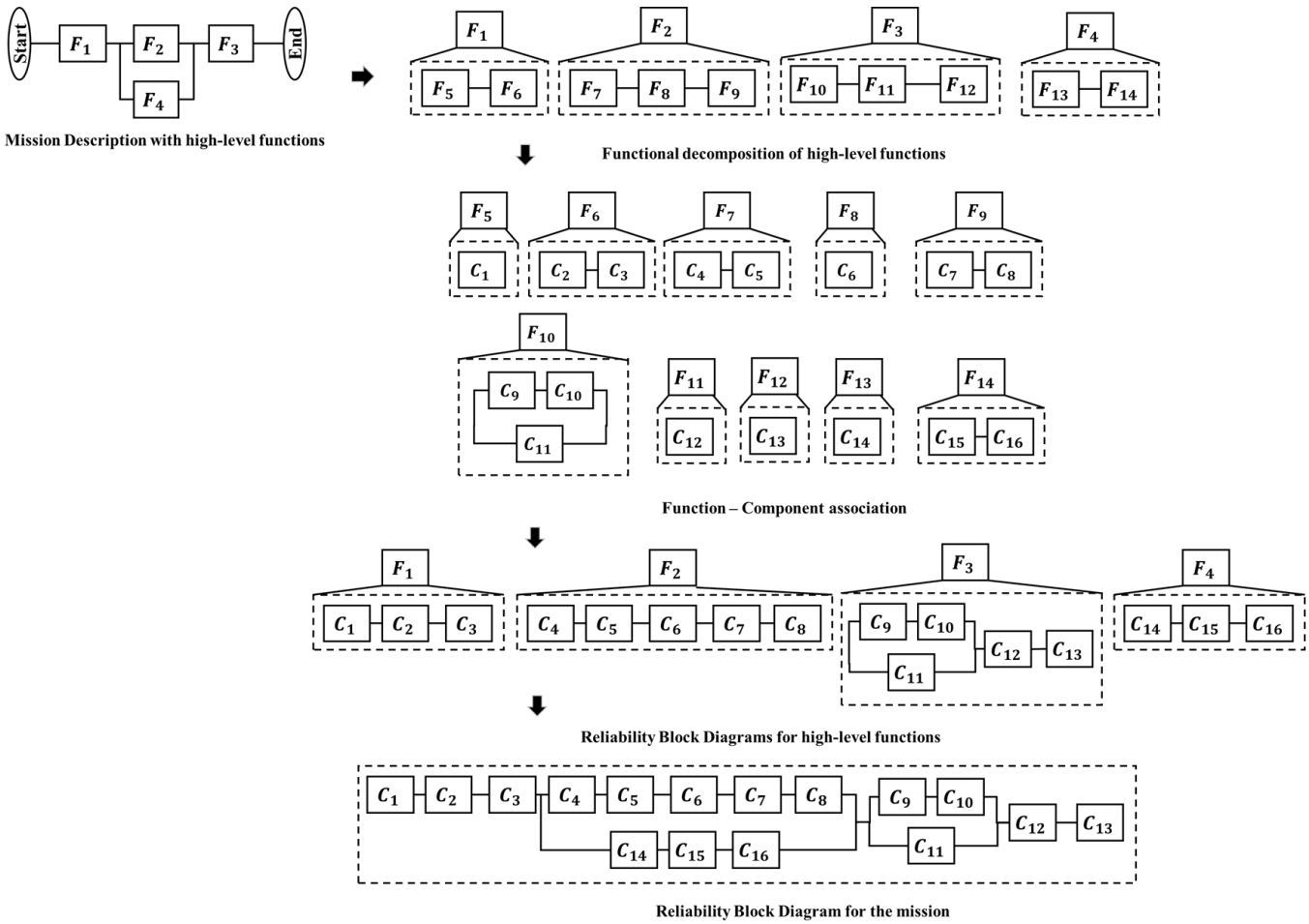


Figure 3. Methodology for Reliability Assessment

5. EXAMPLE: Radio-Controlled Car

Mission Description - The RC Car, which initially is at point A has to move to point B and perform surveillance at point B using a camera mounted on it. The car is amphibious and can move from A to B either on land or in water as shown in Figure 4. Along with the land powertrain, a propeller system is also built-in to the RC Car to move in water. The width of the water body is assumed to be 1.5 mile. The total distance to be covered when moving on land from A to B is 2.5 mile. The speeds when moving on land and in water are assumed to be 7.5 mph and 3 mph respectively. The RC Car as modeled in GME (Ledeczi, Maroti, Bakay, Karsai, Garrett, Thomason & Volgyesi, 2001) is shown in Figure 5. A simple model of the RC Car is used for illustration and therefore has limited capabilities in terms of functions that can be carried out. The RC Car can move forward, backward, turn left and turn right. To stop the car, thrust is to be exerted in the opposite direction of motion i.e., if the car is moving forward then thrust is to be exerted in the reverse direction to stop the car. This forms the primary braking system and along with this, a secondary emergency braking system is also assumed to be available. From the mission description, the function-time plot can be constructed as shown in Figure 6. The mission can be divided into two high-level functions – 1) A function F_{AB} that represents the movement of the RC Car from A to B and 2) a function F_S that represents the surveillance activity at point B. To complete function F_{AB}, the RC Car can choose between two alternate paths – to move on land, represented by F_{AB_L} or in water, represented by F_{AB_W}. The function F_{AB_L} is decomposed into three sub-functions - 1) Moving from A to C, represented by $F_{AB_L}.F_{AC}$ 2) Moving from C to D, represented by $F_{AB_L}.F_{CD}$ 3) Moving from D to B, represented by $F_{AB_L}.F_{DB}$. The locations of points C, D are shown in Figure 4. The successful completion of all these three sub-functions results in the successful completion of function F_{AB_L}. Each of the sub-functions is further decomposed into a number of smaller leaf-level functions and successful completion of all the leaf-level function results in the completion of a sub-function. Table 1 shows the sub-functions of F_{AB_L} and their associated leaf-level functions. In the case of function F_{AB_W}, the function itself is a leaf-level function and therefore cannot be decomposed further. Figure 7 provides the decomposition of the high- level function in moving from A to B (F_{AB}) along with duration of each of the leaf-level functions required.

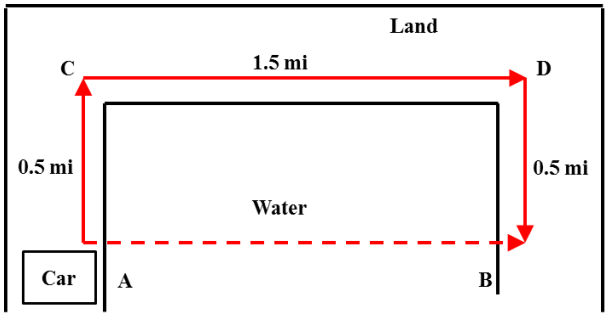


Figure 4. Mission Description

Table 1.Sub-functions of F_{AB_L} and their leaf-level functions

Sub-Function	Leaf-Level Function	Notation
$F_{AB_L}.F_{AC}$	Turn Left at A	F_1
	Move Forward from A to C	F_2
	Turn right at C	F_3
$F_{AB_L}.F_{CD}$	Move forward from C to D	F_4
	Turn right at D	F_5
$F_{AB_L}.F_{DB}$	Move forward from D to B	F_6
	Turn left at B	F_7
	Brake and stop at B	F_8

Using the hierarchical decomposition, the function F_{AB} can be expressed in terms of the leaf-level functions as

$$F_{AB} = \big((F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge F_5 \wedge F_6 \wedge F_7 \wedge F_8) \vee (F_9 \wedge F_8)\big) \quad (5)$$

The next step after obtaining the hierarchical decomposition is to associate component assemblies to carry out each of the atomic-level functions. Table 2 shows the list of component assemblies available in the RC Car system along with their MTTF values and Table 3 shows the association between atomic-level functions and component assemblies. To demonstrate the methodology, MTTF values for the components are assumed. After obtaining the functional decomposition (hierarchical decomposition) and associations between functions and components, the reliability of the overall mission is computed from reliability information of component assemblies through a reliability block diagram. The construction of a reliability block diagram can be carried out in two steps – (1) the atomic functions in Equation 1 are substituted with their associated component assemblies from Table 3, (2) all the components connected with $'\wedge'$ are written in series, whereas components connected with $'\vee'$ are written in parallel. The reliability block diagram for the mission is assembled using the PyEDA package in python.

All the components are assumed to be in the second phase of the bathtub curve where the failure rates are constant and failure probability is modeled using exponential distribution as stated in Section 3.

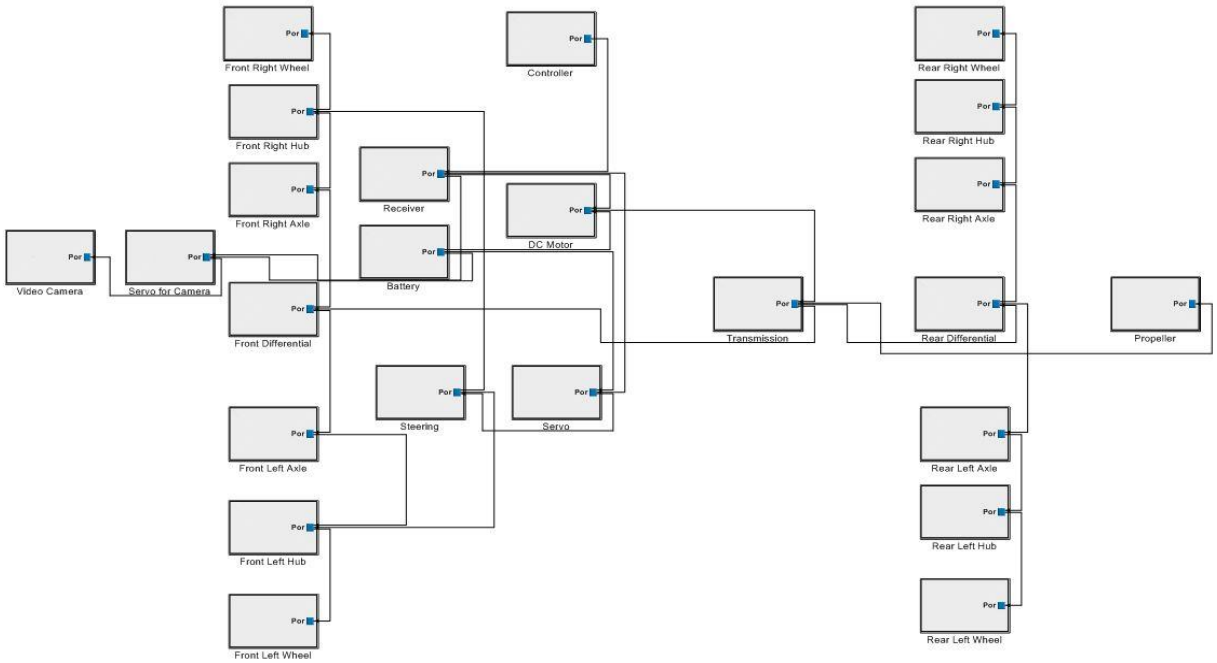


Figure 5. Modeling of the RC Car

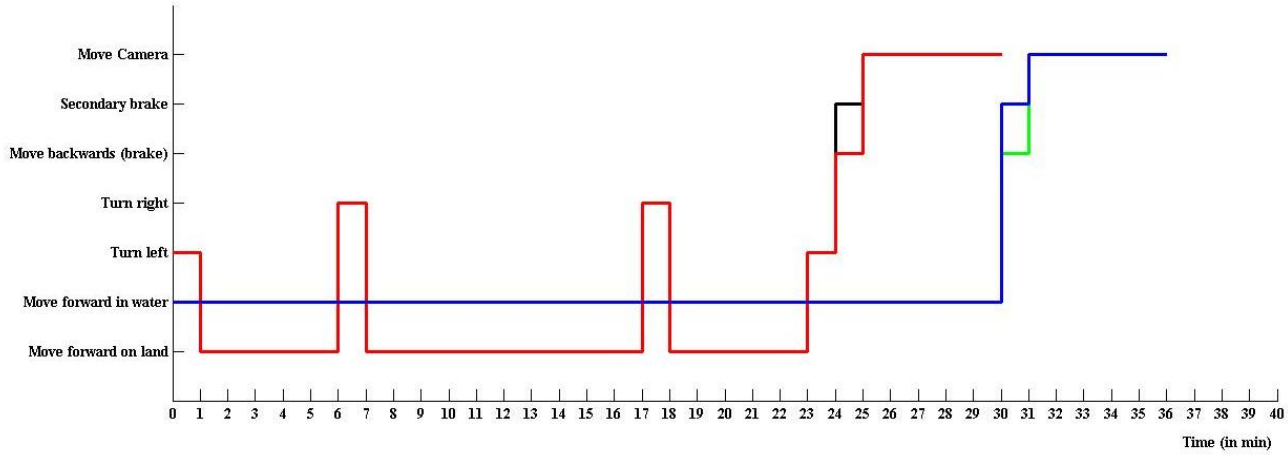


Figure 6. Function-Time Diagram for the mission

The reliability block is constructed using the functional decomposition and function-component association. Using the available MTTF values, the reliability of the mission can be computed as 0.909.

Case 1: Real-time reliability assessment

Assume that the mission was being undertaken by moving in water to reach from A to B. Let T denote the time into the mission, therefore T=0 and T=36 refer to the start and the end of the mission (Figure 6). Tables 4 show the functions required to complete the mission at time T=0 and time T=20.

The third column in Table 4 can be interpreted as follows - At T=20, for successful completion of the mission, F_{AB_W} is required for 10 more minutes (T=20 to T=30), Braking is required for 1 minute and surveillance for 5 minutes. And all

these three functions are required in succession, as shown in the function-time diagram (Figure 6). The reliability block diagram for the mission at time T=20, is assembled using the PyEDA package. Using the reliability block diagram and the MTTF values of the components, the reliability (probability of success) of the remaining portion of mission can be computed.

Case 2: Component unavailability

Assume that at time T = 20, the secondary brake fails and becomes unavailable (due to some unknown reason). Since the braking function has redundancy (primary and secondary), the reliability of the braking function decreases. The reliability of the remaining mission, given that there is no failure up to T = 20, decreases from 0.963 to 0.959.

Case 3: Mission Feasibility

Assume that the camera fails during the travel from A to B in water. Since camera component becomes unavailable, the surveillance cannot be carried out at point B because there is no redundancy available for the surveillance function. Therefore, the mission cannot be carried out successfully. A real-time decision can be made to abort the mission and bring back the RC Car to point A.

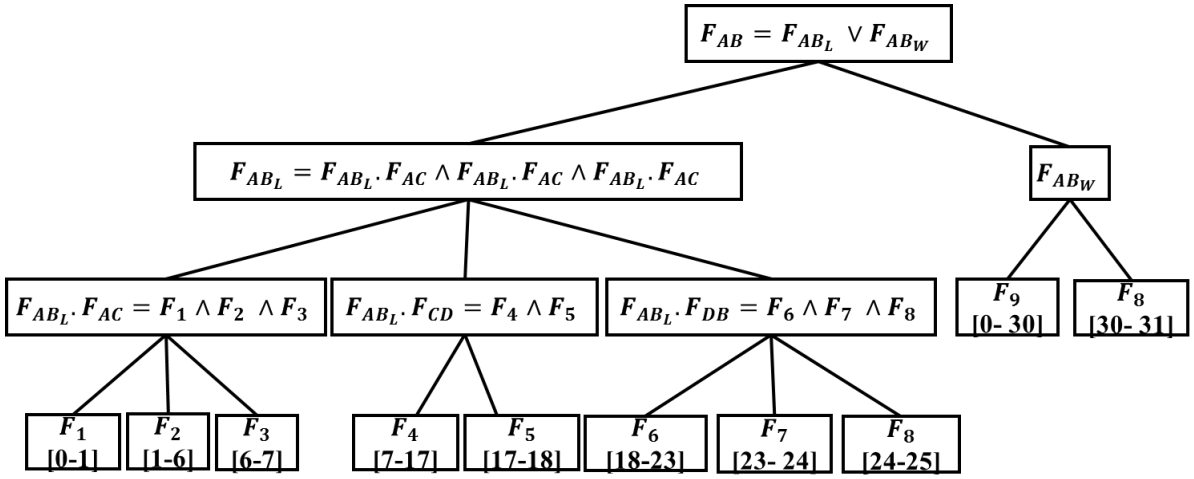


Figure 7. Hierarchical decomposition of the function of moving from A to B (F_{AB})

Table 2. Components in the RC Car and their MTTF values

Component Assembly	Notation	MTTF
Front Wheel System	W_F	5000
Front Hub System	H_F	3000
Front Axle System	A_F	4000
Front Differential	D_F	3000
Transmission	T	2000
DC Motor	DCM	2000
Battery	B	5000
Receiver	R	5000
Servo	S	2000
Steering	St	2000
Servo for Camera	S_C	2000
Camera	C	3000
Rear Differential	D_R	3000
Rear Axle System	A_R	4000
Rear Hub System	H_R	3000
Rear Wheel System	W_R	5000
Propeller	P	700
Chassis	Ch	5000
Secondary Brake System	E_B	1000

Table 3.Leaf-level functions and their components

Function	Component Assembly
F_1, F_3, F_5, F_7	$R \wedge B \wedge S \wedge St \wedge H_F \wedge W_F \wedge Ch$
F_2, F_4, F_6	$R \wedge B \wedge DCM \wedge T \wedge D_F \wedge D_R \wedge A_F \wedge A_R \wedge H_F \wedge H_R \wedge W_F \wedge W_R \wedge Ch$
F_8	$(R \wedge B \wedge DCM \wedge T \wedge D_F \wedge D_R \wedge A_F \wedge A_R \wedge H_F \wedge H_R \wedge W_F \wedge W_R \wedge Ch) \vee (E_B \wedge Ch)$
F_9	$R \wedge B \wedge DCM \wedge T \wedge P \wedge Ch$
F_S	$R \wedge B \wedge S_C \wedge C \wedge Ch$

Table 4. Functions required at T=0 and T=20

Function	Duration required	
	T=0	T=20
Moving in water (F_9)	30	10
Brake at point B (F_8)	1	1
Surveillance (F_S)	5	5

6. CONCLUSION

In this paper, a formal framework has been proposed for reliability assessment of component-based systems, in carrying out specific missions. The key concepts are (1) Functional decomposition, (2) Function-Component association, and (3) Extraction of mission-level reliability diagram. The system undergoing the mission is modeled in Generic Modeling Environment (GME) and each component is associated to the list of functions that it is required for. Functional decomposition is performed for each of the high-level functions in the mission and represented using a hierarchical tree-structure. For each of the leaf-level function, the corresponding components are extracted from the GME and exported to the PyEDA package in Python, where a reliability block diagram is obtained using Boolean expressions. Using the reliability information of the components, the reliability assessment of the mission can be carried out. This procedure can be used for real-time reliability assessment and monitoring of the mission. Using the reliability estimates of the mission as a function of time, real time decisions can be taken such as to continue the mission, abort the mission, perform a simpler mission, or choose a particular path that maximizes the reliability of the mission when there is redundancy available in carrying out functions in a mission. The proposed methodology is demonstrated using a radio-controlled car in carrying out a simple surveillance mission. Future work should address reliability assessment in the presence of dependencies between failures in the components, operational dependencies, and mission dependencies. Also, failure rates that depend on the degradation of the components will need to be considered.

ACKNOWLEDGEMENT

The research presented in this paper was supported by funds from DARPA as part of the Adaptive Vehicle Make (AVM) Project. The support is gratefully acknowledged. The authors also thank Dr. Gautam Biswas and Dr. Xenofon Koutsoukos of Vanderbilt University for valuable discussions during this research.

REFERENCES

Bennetts, R. G. (1982). Analysis of reliability block diagrams by Boolean techniques. *IEEE Transactions on Reliability*, 31(2), 159-166.

Bouti, A., & Kadi, D. A. (1994). A state-of-the-art review of FMEA/FMECA. *International Journal of reliability, quality and safety engineering*, 1(04), 515-543.

Dubey, A., Mahadevan, N., & Karsai, G. (2012). The inertial measurement unit example: A software health management case study. ISIS, 12, 101.

Elsayed, E. A. (2012). *Reliability engineering*. Wiley Publishing.

Ericson, C. A. (2005). Event Tree Analysis. *Hazard Analysis Techniques for System Safety*, 223-234.

Filliben, J. J. (2002). NIST/SEMTECH Engineering Statistics Handbook. Gaithersburg: www. itl. nist. gov/div898/handbook, NIST.

Greenfield, M. A. (2001). NASA's use of quantitative risk assessment for safety upgrades. *Space safety, rescue and quality*, 153-159.

Kececioglu, D. (1972). Reliability analysis of mechanical components and systems. *Nuclear Engineering and Design*, 19(2), 259-290.

Krishnamurthy, S., & Mathur, A. P. (1997). On the estimation of reliability of a software system using reliabilities of its components. *Proceedings of 8th International Symposium in Software Reliability Engineering* (pp. 146-155). IEEE.

Kurtoglu, T., & Tumer, I. Y. (2008). A graph-based fault identification and propagation framework for functional design of complex systems. *Journal of Mechanical Design*, 130, 051401.

Kurtoglu, T., Tumer, I. Y., & Jensen, D. C. (2010). A functional failure reasoning methodology for evaluation of conceptual system architectures. *Research in Engineering Design*, 21(4), 209-234.

Ledeczi, A., Maroti, M., Bakay, A., Karsai, G., Garrett, J., Thomason, C. & Volgyesi, P. (2001). The generic modeling environment. *Workshop on Intelligent Signal Processing*, Budapest, Hungary (Vol. 17).

Lee, W. S., Grosh, D. L., Tillman, F. A., & Lie, C. H. (1985). Fault Tree Analysis, Methods, and Applications. A Review. *IEEE Transactions on Reliability*, 34(3), 194-203.

Mahadevan, N., Dubey, A., Balasubramanian, D., & Karsai, G. (2013). Deliberative, search-based mitigation strategies for model-based software health management. *Innovations in Systems and Software Engineering*, 9(4), 293-318.

Modarres, M. (2008). Probabilistic Risk Assessment. *Handbook of Performability Engineering* (pp. 699-718). Springer London.

Mosterman, P. (2007). Model-based design of embedded systems. *IEEE International Conference on Microelectronic Systems Education*, IEEE.

Phillips, A. M. (2002). Functional decomposition in a vehicle control system. *Proceedings of American Control Conference* (Vol. 5, pp. 3713-3718). IEEE.

Python library for Electronic Design Automation (PyEDA) Documentation [Online]. https://media.readthedocs.org/pdf/pyeda/latest/pyeda.pdf. Last accessed – May 30, 2014

Schattkowsky, T., & Muller, W. (2004). Model-based design of embedded systems. *Proceedings of Seventh IEEE International Symposium on Object-Oriented Real-Time Distributed Computing* (pp. 113-128). IEEE.

Teng, S. H. G., & Ho, S. Y. M. (1996). Failure mode and effects analysis: an integrated approach for product design and process control. *International Journal of Quality & Reliability Management*, 13(5), 8-26.

Wood, A. P. (2001). Reliability-metric varieties and their relationships. *Proceedings of Reliability and Maintainability Symposium* (pp. 110-115). IEEE.

APPENDIX

Definitions

Mission: A mission can be regarded as a time-interval sequence of high-level functions. A mission provides information of all the high-level functions to be carried out at each instant of time. At each time instant, one or more high-level functions can be carried out. The mission is usually represented using a function-time plot.

Functional Decomposition: Functional decomposition is the process of decomposing a high-level function into a set of leaf-level functions (Kurtoglu & Tumer, 2008). A leaf-level function is a function that cannot be decomposed any further. All the leaf-level functions are required for the successful completion of the high-level function. Functional decomposition of a high-level function can be represented using a hierarchical tree-structure. The dependency relationships can be written using the following Boolean relationships – and, or, r-out-of-n. The number of branches in the tree depends on the fidelity of the analysis required. At any instant of time, one or more high-level functions can be happening; therefore one or more dependency trees are active. A leaf-level function might be required for several high-level functions and therefore appears in several trees

Function-Component association: The next step after functional decomposition is association of each leaf-level function to corresponding component or component assemblies (Kurtoglu, Tumer & Jensen, 2010). Again, Boolean relationships are used to represent the association of components to its functions. The Boolean relationships – and, or, r-out-of-n, are used to associate each leaf-level function to its component assembly. A component can provide more than one leaf-level functions but a leaf-level function cannot be associated with more than one component unless the components are the same.

Component availability: Component availability refers to the availability of a component for usage at any time instant during the mission.

Function availability: Function availability refers to the availability of a function for operation. For a function to be available, all the components required for the implementation of this function should be available.

Mission Feasibility: Mission Feasibility refers to the possibility of completion of the mission given the current state of the components. At any instant of time, if all the components are available to carry out all the functions required at later times in the mission, then it can be concluded that the mission is feasible given the current state of the components. If any of the components becomes unavailable and the component is required at a later time, then the corresponding function cannot be carried out. If there are no alternate possibilities available to carry out this function, then this results in the mission being infeasible.

Redundancy: If a function can be carried out even when a component becomes unavailable, then it can be concluded that there is redundancy in the function with respect to that component.

Sensor Fault Diagnosis in Quadrotors Using Nonlinear Adaptive Estimators

Remus C Avram[1], Xiaodong Zhang[2] and Jacob Campbell[3]

[1,2] *Wright State University, Dayton, Ohio, 45404, USA*
avram.3@wright.edu
xioadong.zhang@wright.edu

[3] *Air Force Research Laboratory, Dayton, OH, 45404, USA*
jacob.campbell.3@us.af.mil

ABSTRACT

Unmanned Aerial Vehicles (UAVs) have attracted significant attentions in recent years due to their potentials in various military and civilian applications. Small UAVs are often equipped with low-cost and lightweight micro-electro-mechanical systems (MEMS) inertial measurement units including 3-axis gyro, accelerometer and magnetometer. The measurements provided by gyros and accelerometers often suffer from bias and excessive noise as a result of temperature variations, vibration, etc. This paper presents a sensor fault diagnostic method for quadrotor UAVs. Specifically, we consider the faults in the gyro and accelerometer. A model-based sensor fault detection and isolation (FDI) estimation method is presented. The proposed FDI method adopts the idea that accelerometer and gyroscopic measurements coincide with the translational and rotational forces represented in the UAV dynamics. Thus, the faults in accelerometer and gyroscope can be represented as virtual actuator faults in the quadrotor state equations. Two diagnostic estimators are designed to provide structured FDI residuals allowing simultaneous detection and isolation of gyroscope and accelerometer sensor bias. In addition, nonlinear adaptive estimators are designed to provide an estimate of the unknown sensor bias. The parameter convergence property of the adaptive estimation scheme is analyzed. Simulation studies utilizing a nonlinear quadrotor UAV model are used to illustrate the effectiveness of the proposed method.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have attracted significant attentions in recent years due to their potentials in various military and civilian applications, including security patrol, search and rescue in hazardous environment, surveillance and classification, attack and rendezvous (Shima & Rasmussen, 2008). In addition, compared with manned systems, the reductions in operations and support costs for unmanned vehicles offer the advantage for life cycle cost savings (US Dept. of Defense, 2012). The potential capabilities offered by unmanned vehicles have been well recognized and continue to expand. In manned systems, the human operator functions as the central integrator of the onboard systems to achieve their operational capabilities. Due to the requirement of autonomous operations without a human operator, autonomous control of UAVs is much more challenging. For instance, UAVs currently suffer mishaps at 10 to 100 times the rate incurred by their manned counterparts (US Dept. of Defense, 2012, 2000). In order to enhance the reliability, survivability and autonomy of UAVs, advanced intelligent control and health management technologies are required, which will enable UAVs to have the capabilities of state awareness and self-adaptation (Sharifi, Mirzaei, Gordon, & Zhang, 2010; Vachtsevanos, Tang, Drozeski, & Gutierrez, 2005).

Most quadrotors used in research, are often equipped with low-cost and lightweight micro-electro-mechanical systems (MEMS) inertial measurement units (IMU) including 3-axis gyro, accelerometer and magnetometer. These sensors serve an essential role in most quadrotor control schemes. However, due to their intrinsic components and fabrication process, IMUs are vulnerable to exogenous signals and prone to faults. Specifically, accelerometer and gyroscope measurements are susceptible to bias and excessive noise as a result of temperature variation, vibration, etc. The detection and estimation of accelerometer and gyroscope faults plays a crucial role in the safe operations of quadrotors.

Several researchers have investigated the problem of quadrotor IMU sensor fault diagnosis based on linearized quadrotor dynamic model (Sharifi et al., 2010; Freddi, Longhi, & Monteriú, 2009; Dydek, Annaswamy, & Lavretsky, 2013; Here-

Remus Avram et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

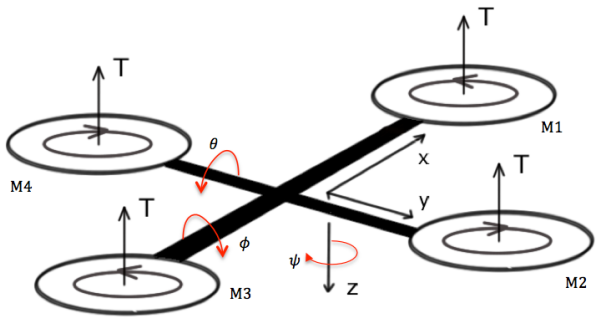


Figure 1. Quadrotor Model in "+" configuration.

dia, Ollero, Mahtani, & Bejar, 2005). A few papers have considered the Luenberger or Kalman filter based observers in order to generate residuals for fault diagnosis purposes (see, for example (Freddi et al., 2009; Heredia et al., 2005; Lantos & Marton, 2011)). These methods rely on linearization of the system around a set of equilibrium points. However, the dynamics of the quadrotor are highly nonlinear and the states can be strongly coupled. In recent years, considerable research effort has been devoted to fault diagnosis of nonlinear systems under various kinds of assumptions and fault scenarios (Blanke, Kinnaert, Lunze, & Staroswiecki, 2005). In this paper we present a nonlinear method for detecting, isolating and estimating sensor bias faults in accelerometer and gyroscope measurements of quadrotor UAVs. Based on the fact that the accelerometer and the gyroscope measure forces/torque acting directly on the UAV body, the quadrotor dynamics are expressed in terms of the IMU sensor measurements. Two diagnostic estimators are designed to provide structured fault detection and isolation (FDI) residuals allowing simultaneous detection and isolation of gyroscope and accelerometer sensor bias. In addition, by utilizing nonlinear adaptive estimation techniques (Zhang, Polycarpou, & Parsini, 2001), adaptive estimators are employed to provide an estimate of the unknown sensor bias. The parameter convergence property of the adaptive estimation scheme is analyzed.

The remainder of the paper is organized as follows. Section II formulates the problem of sensor FDI for quadrotor UAVs. The proposed fault detection and isolation method is presented in Section III. Section IV describes the adaptive estimator algorithms for estimation of sensor bias magnitude and provides conditions for parameter convergence. Section V and VI present simulation results and direction of future research, respectively.

2. PROBLEM FORMULATION

Several works focus on quadrotor modeling see for example (Bramwell, Done, & Balmford, 2001) and (Castillo, Lozano, & Dzul, 2005). More recently, (Pounds, Mahony, & Gresham, 2004; Bangura & Mahony, 2012) have aimed for higher modeling accuracy by including drag force, Coriolis effects, blade flapping effects etc. Accurate modeling plays an important role in quadrotor control, especially in the case of aggressive maneuvers, tight group formations, etc. However, when the quadrotor is in a non-aggressive maneuver state, these effects become very small in comparison to gravitational pull and thrust generated by the rotors. As in (Leishman, Jr., Beard, & McLain, 2014) and (Martin & Salaün, 2010), the dynamic model used in this paper considers the gravity, thrust generated by the rotors and drag forces acting on the quadrotor body. Figure 1 shows a simplifed model of the quadrotor along with the assumed body frame orientation and Euler angles convention using the right-hand rule. The quadrotor nominal system dynamics are derived from the Newton-Euler equations of motion and are given by:

$$\dot{p}_E = v_E \tag{1}$$

$$\dot{v}_E = \frac{1}{m} R_{EB}(\eta) \left(\begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} - c_{\mathbf{d}} v_B \right) + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2}$$

$$\dot{\eta} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \omega \tag{3}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{bmatrix} \tag{4}$$

where $p_E \in \mathbb{R}^3$ is the inertial position, $v_E \in \mathbb{R}^3$ is the velocity expressed in the Earth frame, $\eta = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ are the roll, pitch and yaw Euler angles, respectively, and $\omega = [p\, q\, r]^T$ represents the angular rates, m is the mass of the quadrotor, and g is the gravitational acceleration. The terms J_x, J_y and J_z represent the quadrotor inertias about the body x-, y- and z-axis, respectively. Note that the quadrotor is assumed to be symmetric about the xz and yz planes (i.e. the product of inertias is zero). T represents the total thrust generated by the rotors, τ_ϕ, τ_θ, τ_ψ are the torques acting on the quadrotor around the body x-, y- and z-axis, respectively. The term $c_{\mathbf{d}} v_B$ represents the drag force acting on the vehicle frame, with $c_{\mathbf{d}}$ being drag force coefficient and v_B is the velocity of the UAV relative to the body frame.

The system model described by Eq (1) - (4) is expressed with the velocity relative to the inertial frame. The inertial coordinate system is assumed to have the positive x-axis pointing North, the positive y-axis pointing East and positive z-axis pointing down towards the Earth's center. The transformation from the body frame to inertial frame is given by the rotation matrix R_{EB} and is defined based on a 3-2-1 rotation sequence as follows:

$$R_{EB}(\eta) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

where $s\cdot$ and $c\cdot$ are short hand notations for the $\sin(\cdot)$ and $\cos(\cdot)$ functions, respectively.

MEMS sensors, such as accelerometers and gyroscopes, measure forces and moments acting in the body frame. The quantity expressed inside the parenthesis in the inertial velocity Eq. (2), represents all the forces acting on the body. Therefore, the velocity dynamic equation can be adjusted to reflect accelerometer measurements. Similarly, the evolution of Euler angles can be rewritten in terms of gyroscope measurements (Leishman et al., 2014; Ireland & Anderson, 2012). By considering IMU measurement susceptibility to a constant bias drift, the accelerometer and gyroscope sensor measurements are given by:

$$y_a = a + b_a = \frac{1}{m}\left(\begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} - c_{\mathbf{d}} v_B\right) + b_a \tag{5}$$

$$y_\omega = \omega + b_\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + b_\omega \tag{6}$$

where $y_a \in \mathbb{R}^3$ and $y_\omega \in \mathbb{R}^3$ are the measured accelerometer and gyro quantities, respectively, $b_a \in \mathbb{R}^3$ and $b_\omega \in \mathbb{R}^3$ are the possible constant bias in accelerometer and gyroscope measurements, respectively, and a represents the nominal acceleration measurement without bias, that is:

$$a = \frac{1}{m}\left(\begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} - c_{\mathbf{d}} v_B\right) \tag{7}$$

In addition, as in (Ireland & Anderson, 2012) and (Lantos & Marton, 2011), it is assumed that the position measurements in the Earth frame and Euler angles measurements are available. For instance, these measurements can be generated by a camera-based motion capture system, a technology commonly employed for in-door UAV flight (Guenard, Hamel, & Mahony, 2008). Hence, the system model can be augmented by the following output equations:

$$y_p = p_E \tag{8}$$

$$y_\eta = \eta \tag{9}$$

The objective of this research focuses on the detection, isolation and estimation of sensor bias in accelerometer and gyroscope measurements.

3. FAULT DETECTION AND ISOLATION

This section presents the proposed method for detecting and isolating sensor faults in accelerometer and gyroscope measurements. Substituting the sensor model from Eq (5)-(6) into the systems dynamics Eq (1)-(4), we obtain:

$$\dot{p}_E = v_E \tag{10}$$

$$\dot{v}_E = R_{EB}(\eta) y_a + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - R_{EB}(\eta) b_a \tag{11}$$

$$\dot{\eta} = T(\eta) y_\omega - T(\eta) b_\omega \tag{12}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{bmatrix} \tag{13}$$

where $T(\eta)$ is the rotation matrix relating angular rates to Euler angle rates and is given by:

$$T(\eta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}.$$

In order to eliminate the coupling between translational velocity and angular rates when the quadrotor dynamics are represented with velocity relative to the body frame, the quadrotor dynamics are expressed with velocity relative to the earth frame. As can be seen from Eq (10)-(13), a bias in accelerometer measurements affects only the position and velocity states. Conversely, gyroscope measurements affect only Euler angles and angular rates states. Based on this observation, it follows naturally to also divide the fault diagnosis of these two sensor faults. The proposed fault detection, isolation and estimation architecture is shown in Figure 2. As can be seen, two FDI estimators monitor the system for fault occurrences in accelerometer and gyroscope measurements. Once a fault is detected and isolated, the corresponding nonlinear adaptive estimator is activated for sensor bias estimation purposes.

3.1. Gyroscope Bias Diagnostic Estimator

As can be seen from the dynamics of the quadrotor, given by equations (10)-(13), the bias in the gyroscope measurements only affects the attitude and rotation dynamics given by Eq (12)-(13). Since the attitude angles given by the state vector η are assumed to be measurable (see Section 2), based on Eq (12)-(13) and adaptive estimation schemes, such as the series-parallel model (Ioannou & Sun, 1996), the fault diagnostic estimator for the gyroscope bias can be designed as follows:

$$\dot{\hat{\eta}} = -\Lambda(\hat{\eta} - \eta) + T(\eta) y_\omega \tag{14}$$

where $\hat{\eta} \in \mathbb{R}^3$ are the Euler angle estimates, $\Lambda \in \mathbb{R}^{3\times3}$ and $\Gamma \in \mathbb{R}^{3\times3}$ are positive-definite, diagonal design matrices. Let the Euler angle estimation error be defined as:

$$\tilde{\eta} \triangleq \eta - \hat{\eta} \tag{15}$$

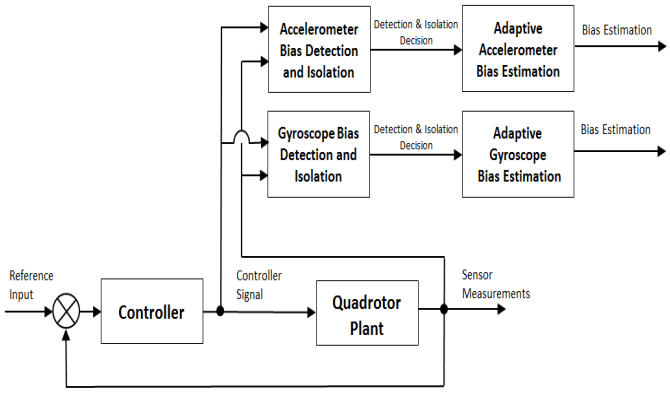


Figure 2. Fault Detection, Isolation and Estimation Architecture.

Then, based on Eq (12) and Eq (14), we have:

$$\dot{\tilde{\eta}} = \dot{\eta} - \dot{\hat{\eta}} = -\Lambda\tilde{\eta} - T(\eta)b_\omega \tag{16}$$

Equation (16) guarantees that the Euler angles estimation error converges asymptotically to zero in the absence of gyroscope sensor bias. In addition, in the presence of a non-zero bias b_ω, based on Eq (16) it can be seen that the residual $\tilde{\eta}$ will deviate from zero. Therefore, if any component of the state estimation error $\tilde{\eta}$ is significantly different from zero, we can conclude that a fault in the gyroscope measurements has occurred.

3.2. Accelerometer Bias Diagnostic Estimator

The dynamics of UAV position and velocity relative to the inertial frame given by Eq (10) and Eq (11) can be put in the following state space model:

$$\begin{aligned} \dot{x} &= Ax + f(\eta, y_a) + G_a(\eta)b_a \\ y &= Cx \end{aligned} \tag{17}$$

where $x = [p_E^T \; v_E^T]^T$, $y = p_E$, and

$$A = \begin{bmatrix} 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix},$$

$$G_a(\eta) = \begin{bmatrix} 0_{3\times3} \\ -R_{EB} \end{bmatrix},$$

$$f(\eta, y_a) = \begin{bmatrix} 0_{3\times1} \\ R_{EB}y_a + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \end{bmatrix}$$

and $C = [I_3, \; 0_{3\times3}]$, where I_3 is a 3×3 identity matrix, $0_{3\times3}$ is a 3×3 matrix with all entries zero and $0_{3\times1}$ is a 3×1 zero vector. Based on this configuration, the following fault diagnostic observer is chosen :

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + f(\eta, y_a) + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \tag{18}$$

where $\hat{x} \in \mathbb{R}^6$ represents the inertial position and velocity estimation, $\hat{y} \in \mathbb{R}^3$ are the predicted position outputs, and L is a design matrix chosen such that the matrix $\bar{A} \triangleq (A - LC)$ is stable. From the definition of matrices A and C given by Eq (17) it is straightforward to show that the system is observable. Therfore, the matrix L can be easily designed. Defining the state estimation error as: $\tilde{x} \triangleq x - \hat{x}$ and the quadrotor position estimation error as $\tilde{y} \triangleq y - \hat{y}$, it follows that:

$$\begin{aligned} \dot{\tilde{x}} &= \bar{A}\tilde{x} + G_a(\eta)b_a \\ \tilde{y} &= C\tilde{x} \; . \end{aligned} \tag{19}$$

Clearly, the output estimation error $\tilde{y}$ reaches zero asymptotically in the absence of the accelerometer bias b_a. Furthermore it can be seen from Eq (19) the residual $\tilde{y}$ is only sensitive to the bias b_a. Therefore, if any component of the position estimation error $\tilde{y}$ deviates significantly from zero, we can conclude that a fault in the accelerometer sensor measurement has occurred.

3.3. Fault Detection and Isolation Decision Scheme

As described in section 3.1 and 3.2, the two fault diagnostic estimators are designed such that each of them is only sensitive to one type of sensor faults. Based on this observation, the residuals $\tilde{\eta}$ and $\tilde{y}$ generated by Eq (16) and Eq (19) can also be used as structured residuals for fault isolation. More specifically, we have the following fault detection and isolation decision scheme:

- In the absence of any faults, all components of the residuals $\tilde{\eta}$ and $\tilde{y}$ should be close to zero.
- If all components of the residual $\tilde{\eta}$ remain around zero, and at least one component of the residual $\tilde{y}$ is significantly different from zero, then we conclude that an accelerometer fault has occurred.
- If all components of the residual $\tilde{y}$ remain around zero, and at least one component of the residual $\tilde{\eta}$ is significantly different from zero, then we conclude that a gyroscope fault has occurred.
- If at least one component of the residual $\tilde{\eta}$ is significantly different from zero, and at least one component of the residuals $\tilde{y}$ is significantly different from zero, then we conclude that both a gyroscope and accelerometer sensor measurement fault has occurred.

The above FDI decision scheme is summarized in Table 1, where "0" represents nearly zero residuals, and "1" represents significantly large residuals.

Table 1. Fault Isolation Decision Truth Table.

	No Fault	Gyro Bias	Accel Bias	Accel & Gyro Bias
$\tilde{\eta}$	0	0	1	1
$\tilde{y}$	0	1	0	1

4. FAULT ESTIMATION

After a sensor fault is detected and isolated, it is also crucial to provide an estimation of the sensor bias to improve the performance of the closed loop control system. As shown in Figure 2, once a fault has been detected and isolated, the corresponding nonlinear adaptive bias estimator is activated with the purpose of estimating the fault magnitude in the accelerometer and/or gyroscope measurements. In this section, we describe the design of nonlinear adaptive estimators for sensor bias estimation.

4.1. Accelerometer Fault Estimation

Based on Eq (17), the adaptive observer for estimating the accelerometer bias magnitude is chosen as:

$$\dot{\hat{x}} = A\hat{x} + f(\eta, y_a) + L(y - \hat{y}) + G_a(\eta)\hat{b}_a + \Omega\dot{\hat{b}}_a \quad (20)$$

$$\dot{\Omega} = (A - LC)\Omega + G_a(\eta) \quad (21)$$

$$\hat{y} = C\hat{x} \quad (22)$$

where $\hat{x}$ is the estimated position and velocity vector, $\hat{y}$ is the estimated position output, $\hat{b}_a$ is the estimated sensor bias, and L is the observer gain matrix. The adaptation in the above adaptive estimator arises due to the unknown bias b_a. The adaptive law for updating $\hat{b}_a$ is derived using Lyapunov synthesis approach (Ioannou & Sun, 1996; Zhang, 2011). Specifically, the adaptive algorithm is given by:

$$\dot{\hat{b}}_a = \Gamma\Omega^T C^T \tilde{y}_a \quad (23)$$

where $\Gamma > 0$ is a symmetric and positive-definite learning rate matrix, and $\tilde{y}_a \triangleq y_a - \hat{y}_a$ is the output estimation error. Let us also define the state estimation error $\tilde{x} \triangleq x - \hat{x}$. Then, based on Eq (17) and Eq (20), the dynamics governing the state estimation error are given by:

$$\dot{\tilde{x}} = \bar{A}\tilde{x} - G_a(\phi, \theta, \psi)\tilde{b}_a - \Omega\dot{\tilde{b}}_a \quad (24)$$

where $\bar{A} \triangleq A - LC$ and $\tilde{b}_a \triangleq \hat{b}_a - b_a$ is the parameter estimation error. By substituting $G_a(\eta) = \dot{\Omega} - (A - LC)\Omega$ (see Eq (21)) into Eq (24), we have

$$\begin{aligned} \dot{\tilde{x}} &= \bar{A}\tilde{x} - (\dot{\Omega} - \bar{A}\Omega)\tilde{b}_a - \Omega\dot{\tilde{b}}_a \\ &= \bar{A}(\tilde{x} + \Omega\tilde{b}_a) - \dot{\Omega}\tilde{b}_a - \Omega\dot{\tilde{b}}_a \ . \end{aligned} \quad (25)$$

By defining $\bar{x} \triangleq \tilde{x} + \Omega\tilde{b}_a$, the above equation can be rewritten as

$$\dot{\bar{x}} = \bar{A}\bar{x} \ . \quad (26)$$

In addition, the adaptive parameter estimation algorithm (see Eq (23)) can be rewritten as:

$$\begin{aligned} \dot{\hat{b}}_a &= \Gamma\Omega^T C^T \tilde{y}_a \\ &= \Gamma\Omega^T C^T C\tilde{x} \\ &= \Gamma\Omega^T C^T C(\bar{x} - \Omega\tilde{b}_a). \end{aligned} \quad (27)$$

Because the bias b_a is constant, we have $\dot{\tilde{b}}_a = \dot{\hat{b}}_a$. Thus, Eq (27) can be rewritten as:

$$\dot{\tilde{b}}_a = \Gamma\Omega^T C^T C\bar{x} - \Gamma\Omega^T C^T C\Omega\tilde{b}_a. \quad (28)$$

Based on Eq (26), we know $\bar{x}$ converges asymptotically to zero, since $\bar{A}$ is stable by design. In addition, if there exists constants $\alpha_0 > 0$, $T_0 > 0$ and α_1 such that the following condition is satisfied:

$$\alpha_1 I \geq \frac{1}{T_0}\int_t^{t+T_0} \Omega^T C^T C\Omega d\tau \geq \alpha_0 I \quad (29)$$

then we can conclude that the $\tilde{b}_a$ will converge to zero, that is $\hat{b}_a$ converges to the actual value b_a. It is worth noting that the condition given by Eq (29) provides the required persistence of excitation for parameter convergence (Ioannou & Sun, 1996). The nature of UAV flight provides vibrations in practical applications, which may lead to adequate levels of excitation. In addition, this condition can be satisfied by commanding the UAV to perform certain maneuvers.

4.2. Gyroscope Fault Estimation

Based on Eq (12), after the presence of a gyroscope bias fault is detected, the following adaptive estimator is activated in order to estimate the bias in the gyroscope sensor:

$$\dot{\hat{\eta}} = -\Lambda(\hat{\eta} - \eta) + T(\eta)y_\omega - T(\eta)\hat{b}_\omega \quad (30)$$

$$\dot{\hat{b}}_\omega = \Gamma T(\eta)(\eta - \hat{\eta}) \quad (31)$$

where $\hat{\eta}$ is the Euler angle estimate, $\hat{b}_\omega$ represents the estimation of the sensor bias, Λ and Γ are positive definite design matrices. The adaptive law for estimating the bias in gyroscope measurements in Eq (31) is derived using Lyapunov synthesis approach (Ioannou & Sun, 1996). The adaptive scheme in Eq. (30) ensures that the attitude angle estimation error $\tilde{\eta} \triangleq \eta - \hat{\eta}$ converges asymptotically to zero. In addition, in order to ensure parameter convergence, $T(\eta)$ will also have to satisfy the persistence of excitation condition (Ioannou & Sun, 1996), that is:

$$\alpha_1 I \geq \frac{1}{T_0}\int_t^{t+T_0} T(\eta)^T T(\eta) d\tau \geq \alpha_0 I \quad (32)$$

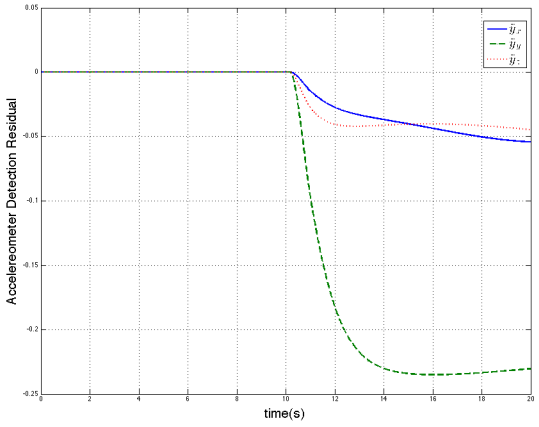
for some constants $\alpha_1 \geq \alpha_0 > 0$ and $T_0 > 0$ and for all $t \geq 0$. Again, we note that vibration present in UAV flight may offer adequate excitation. Additionally, the UAV can be commanded to perform certain maneuvers in order to reach the required levels of persistence of excitation.

5. SIMULATION RESULTS

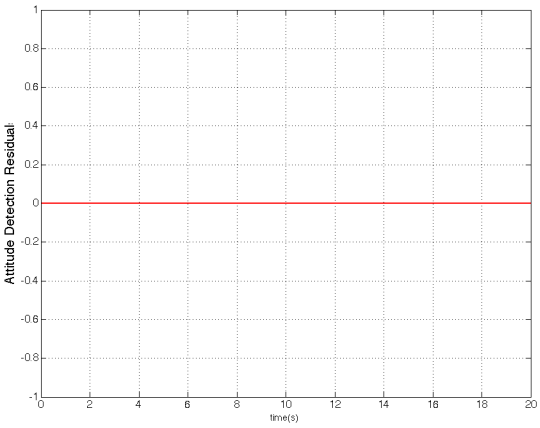
In this section, we present some simulation results in order to illustrate the effectiveness of the proposed sensor bias diagnosis method. Specifically, two cases are studied while the quadrotor is commanded to move along a circular trajectory with a radius of 4 meters for a period of 20 seconds. We control the position and yaw rate of the quadrotor by means of state feedback using a linear quadratic regulator. As previously shown, the fault diagnosis technique employed in this approach is independent of the structure of the controller. Therefore, for brevity, the discussion on the control design is purposely omitted.

The first case studied corresponds to a bias drift in accelerometer measurements. Specifically, at time $t = 10s$ we injected a constant bias of $b_a = [0.2,\ 0.1,\ 0.5]^T m/s^2$ in the accelerometer measurements. Figure 3 shows the FDI residuals generated by the two diagnostic estimators described by Eq (14) and Eq (18), respectively. As can be seen, the components of the residual generated by the estimator corresponding to the accelerometer bias fault become nonzero shortly after fault occurrence, while all residual components generated by the gyroscope fault diagnostic estimator remain zero. Based on the detection and isolation logic given in Table 1, we can conclude that a fault has occurred in the accelerometer measurement. In addition, Figure 4 shows the estimation of the bias in the accelerometer for each axis, respectively, provided by the adaptive estimator (see Eq (23)). As can be seen, the bias estimate correctly reaches the actual bias values in the accelerometer measurements.

The second case corresponds to a bias in the gyroscope measurements injected at time $t = 10s$. The bias magnitude considered is given by $b_\omega = [10°,\ 5°,\ 1°]^T$. Figure 5 shows the time behaviors of the residuals generated by the two diagnositc estimators. As can be seen, the FDI residuals generated by the diagnostic estimator corresponding to the gyroscope fault become nonzero shortly after fault occurrence, and the residuals generated by the estimate corresponding to accelerometer fault always remain zero. Therefore, based on the detection and isolation decision logic given in Table 1, we can conclude that a fault has occurred in gyroscope measurements. Figure 6 shows the estimate of the bias in gyroscope roll, pitch and yaw measurements. As it can be seen, the bias estimate reaches the actual values of the bias in the sensor.



(a) Residuals generated by the diagnostic estimator corresponding to accelerometer bias



(b) Residuals generated by the diagnostic estimator corresponding to gyroscope bias

Figure 3. Fault detection and isolation of accelerometer bias.

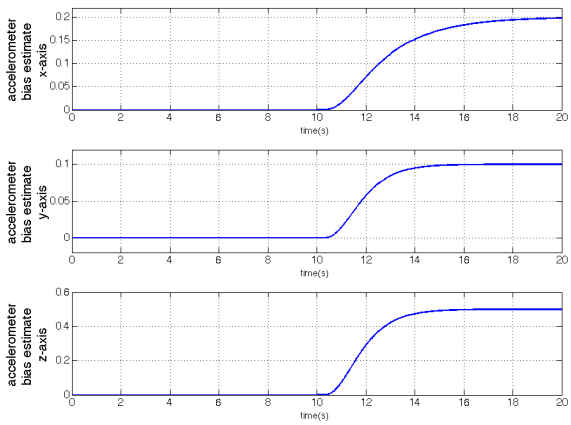
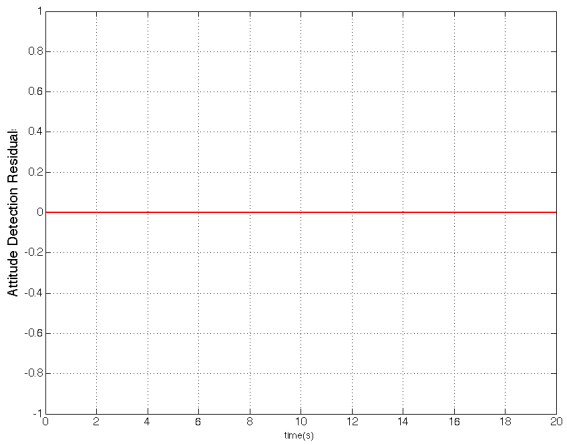
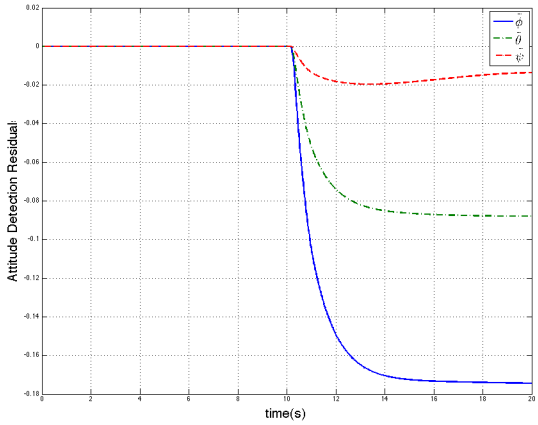


Figure 4. Estimation of accelerometer bias



(a) Residuals generated by the diagnostic estimator corresponding to accelerometer bias



(b) Residuals generated by the diagnostic estimator corresponding to accelerometer bias

Figure 5. Fault detection and isolation of gyroscope bias.

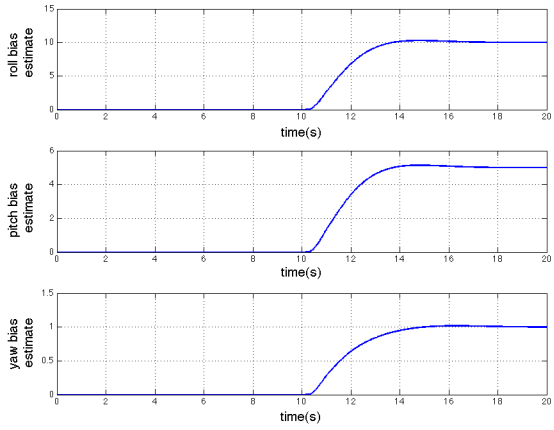


Figure 6. Estimation of gyroscope bias

6. CONCLUSION AND FUTURE WORK

In this paper, we present the design of a nonlinear fault diagnostic method for sensor bias faults in accelerometer and gyroscope measurements of quadrotor UAVs. Based on the idea that accelerometer and gyroscope measurements coincide with translational and rotational forces acting on the body, respectively, two FDI estimators are designed to generate structured residuals for fault detection and isolation. In addition, nonlinear adaptive estimation estimation schemes are presented to provide an estimate of the sensor bias. The effectiveness of the proposed method is illustrated through simulation examples.

In this paper we assumed that Euler angles are available directly for FDI design (for instance, from a motion capture camera system). In some real-world applications, this assumption may not be satisfied. Therefore, the consideration of attitude angle estimation as well as investigation of actuator faults is a direction for future research. In addition, further evaluation of the sensor bias fault diagnostic method through experimental studies with noisy measurements will be conducted.

REFERENCES

Bangura, M., & Mahony, R. (2012). Nonlinear dynamic modelling for high performance control of a quadrotor. In *Proceedings of Austrasian Conference on Robotics and Automation.*

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2005). *Diagnosis and Fault-Tolerant Control*. Springer.

Bramwell, A., Done, G., & Balmford, D. (2001). *Bramwell's Helicopter Dynamics*. Oxford: Butterworth-Heinemann.

Castillo, P., Lozano, R., & Dzul, A. (2005). *Modelling and Control of Mini-Flying Machines*. Springer-Verlag.

Dydek, Z. T., Annaswamy, A. M., & Lavretsky, E. (2013). Adaptive control of quadrotors uavs: A design trade study with flight evaluations. *IEEE Transaction on Automatic Control Systems Technology*, *21*(4).

Freddi, A., Longhi, S., & Monteriú, A. (2009). A model-based fault diagnosis system for a mini-quadrotor. In *7th Workshop on Advanced Control and Diagnosis.*

Guenard, N., Hamel, T., & Mahony, R. (2008). A practical visual servo control for an unmanned aerial vehicle. *IEEE Transaction on Robotics*, *24*(2).

Heredia, G., Ollero, A., Mahtani, R., & Bejar, M. (2005). Detection of sensor faults in autonomous helicopters. In *International Conference on Robotics and Automation.*

Ioannou, P. A., & Sun, J. (1996). *Robust Adaptive Control*. Dover Publications, Inc.

Ireland, M., & Anderson, D. (2012). Development of navi-

gation algorithms for NAP-of-the-earth UAV flight in a constrained urban environment. In *28th International Congress of the Aeronautical Sciences.*

Lantos, B., & Marton, L. (2011). Nonlinear Control of Vehicles and Robots. In (chap. Nonlinear Control of Airplanes and Helicopters). Springer-London.

Leishman, R. C., Jr., J. C. M., Beard, R. W., & McLain, T. (2014). Quadrotors and accelerometers. state estimation with an improved dynamic model. *IEEE Control Systems Magazine*, *34*(1).

Martin, P., & Salaün, E. (2010). The true role of accelerometer feedback in quadrotor control. In *IEEE International Conference on Robotics and Automtation.*

Pounds, P., Mahony, R., & Gresham, J. (2004). Towards dynamically-favourable quad-rotor aerial robots. In *Australasian Conference on Robotics and Automation, ACRA.*

Sharifi, F., Mirzaei, M., Gordon, B. W., & Zhang, Y. (2010). Fault tolerant control of a quadrotor UAV using sliding mode control. In *2010 Conference on Control and Fault Tolerant Systems.*

Shima, T., & Rasmussen, S. (2008). Uav cooperative decision and control: Challenges and practical approaches. In *SIAM.*

US Dept. of Defense. (2000). *Unmanned systems integrated roadmap,FY2000-2025* (Tech. Rep.). Secretary of Defense, Washington, D.C.

US Dept. of Defense. (2012). *Unmanned systems integrated roadmap FY2011-2036* (Tech. Rep.). Secretary of Defense, Washington, D.C.

Vachtsevanos, G., Tang, L., Drozeski, G., & Gutierrez, L. (2005). From mission planning to flight ocntrol of unmanned aerial vehicles: Strategies and implementation tools. *Anual Reviews in Control*, *29*, 101-115.

Zhang, X. (2011). Sensor bias fault detection and isolation in a clas of nonlinear uncertain systems using adaptive estimation. *IEEE Transaction on Automatic Control*, *56*(5).

Zhang, X., Polycarpou, M., & Parsini, T. (2001). Robust fault isolation for a class of non-linear input-output systems. *International Journal Control*, *74*(13).

BIOGRAPHIES

Remus C. Avram is a PhD candidate at Wright State Univerisity, Dayton OH. Remus, received his B.S. in Electrical Engineering with concetration in Computer Engineering at University of Texas, San Antonio, Texas in 2009. In 2011, Remus obtained his M.S. in Electrical Engineering with concetration in Control Systems from Wright Satte University, Dayton, Ohio. He is currently pursuing a doctoral degree in Electrical Engineering. Remus's main research include nonlinear health diagnostics and real-time intelligent control.

Xiaodong Zhang received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from University of Cincinnati, Cincinnati, OH, USA, all in electrical engineering, in 1994, 1997 and 2001, respectively. He is currently an Associate Professor of Electrical Engineering Department, Wright State University, Dayton, OH. His research interests include intelligent control systems, fault diagnosis and prognosis, fault-tolerant control, verification and validation of control systems for safety assurance, etc. He is an Associate Editor of the IEEE Transactions on Control Systems Technology and a member of the IFAC SAFEPROCESS Technical Committee.

Jacob Campbell is a senior electronics engineer in the Air Force Research Laboratory, Reference Systems Branch at Wright-Patterson AFB, Ohio. He received his Ph.D. in electrical engineering from Ohio University. Currently, he manages several Air Force navigation efforts, is the technical representative for DARPAs cold atom inertial research, and serves as team lead for automated aerial refueling navigation technologies. His research interests include navigation in GPS-degraded environments.

Using Johnson Distribution for Automatic Threshold Setting in Wind Turbine Condition Monitoring System

Kun S. Marhadi[1] and Georgios Alexandros Skrimpas[2]

[1,2] *Brüel & Kjær Vibro A/S, 2850 Nærum, Denmark*
kun.marhadi@bkvibro.com
alexandros.skrimpas@bkvibro.com

ABSTRACT

Setting optimal alarm thresholds in vibration based condition monitoring system is inherently difficult. There are no established thresholds for many vibration based measurements. Most of the time, the thresholds are set based on statistics of the collected data available. Often times the underlying probability distribution that describes the data is not known. Choosing an incorrect distribution to describe the data and then setting up thresholds based on the chosen distribution could result in sub-optimal thresholds. Moreover, in wind turbine applications the collected data available may not represent the whole operating conditions of a turbine, which results in uncertainty in the parameters of the fitted probability distribution and the thresholds calculated. In this study Johnson distribution is used to identify shape, location, and scale parameters of distribution that can best fit vibration data. This study shows that using Johnson distribution can eliminate testing or fitting various distributions to the data, and have more direct approach to obtain optimal thresholds. To quantify uncertainty in the thresholds due to limited data, implementations with bootstrap method and Bayesian inference are investigated.

1. INTRODUCTION

Wind turbines are generally subject to aleatory uncertainty due to stochastic nature of the weather and the wind itself. In addition to the stochastic nature that a turbine may experience under normal condition (not experiencing any faults), the varying loads that a wind turbine experience makes monitoring its condition inherently challenging. However, having a condition monitoring system (CMS) dedicated to wind turbines is vital for an effective maintenance program. Such program can help ensure maximum uptime of the machine by minimizing downtime. An example of such system has been demonstrated by (Andersson, Gutt, & Hastings, 2007).

Kun Marhadi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Most CMS for wind turbine applications are based on vibration as described by (Tavner, 2012) and (Crabtree, 2011). A case study of using vibration monitoring to detect and diagnose a fault in the generator bearing of a wind turbine in a real industrial application has also been presented by(Marhadi & Hilmisson, 2013).

As explained by (Marhadi & Hilmisson, 2013), primary components monitored in wind turbines (for vibration based CMS) are the generator, gearbox, main bearings, and tower. Usually accelerometers are installed on these components, and there could be up to 10 accelerometers installed in a wind turbine. The data acquisition unit in a wind turbine usually collects vibration data continuously from each sensor. Different vibration measurements are considered in monitoring different components of a wind turbine. To monitor generator bearings for example, several measurements are used in different frequency ranges. The overall vibration RMS level, ISO RMS [10 - 1000 Hz], high frequency band pass (HFBP [1k - 10k Hz]), high frequency crest factor (HFCF), and several harmonics or orders of the running speed of the generator (e.g. 1X, 2X) are computed by the data acquisition unit continuously from each sensor. Depending on different failure modes or types of fault, there could be more measurements needed and computed from a sensor. To detect gear related problems in a gearbox for example, the tooth/gear mesh frequencies and sideband levels are usually computed in addition to other broad band measurements such as the ISO RMS. The obtained scalar data are usually trended over time. When the trend from a specific measurement (e.g. HFBP or ISO RMS) crosses over a predefined threshold or limit, it will trigger an alarm or warning. Thus it is very important to set the thresholds correctly in order to minimize the number of false alarms.

Given there could be up to 10 sensors installed in a turbine and the number of measurements computed from an individual sensor could vary from 3 to more than 10, the number of thresholds that needs to be set up is consequently large. It is impractical to set them manually. Considering that each wind

turbine is unique like an individual, it is necessary to set the thresholds uniquely to each turbine. It will be even very inefficient if there are thousands of turbines with CMS whose thresholds need to be set manually. More importantly, setting a threshold is often a trade off between missing real alarms due to a fault development and having false alarms. Thus it is important to be able to set the thresholds at the optimum levels automatically with minimum number of adjustments over time.

(Marhadi & Hilmisson, 2013) explained that some limits are determined based on statistics. It is often based on the assumption that the distribution of a vibration measurement follows the Normal (Gaussian) distribution. (Jablonski, Barszcz, Bielecka, & Breuhaus, 2013) discussed a methodology for automatic threshold calculation in a large monitoring system, including a wind turbine application. (Jablonski et al., 2013) showed that different data types or vibration measurements could have significantly different probability distributions other than Gaussian. They investigated several distributions and their comparison in fitting various data types for threshold calculation. (Bechhoefer & Bernhard, 2005) have also presented a case where Gaussian distribution is not appropriate to describe the probability distribution of first order magnitude (1X) of a helicopter shaft. They further explained that it is important that the underlying distribution of a measurement is correct so that the threshold can be determined based on low probability of false alarm.

Earlier work to determine alarm threshold has been presented by (Cempel, 1990), where he investigated the thresholds estimation based on Chebyshev's inequality, Weibull and Pareto distributions. The work also showed its possible application in prognosis although it is more complicated, such as what (Cempel, 1987) showed. Later (Bechhoefer & Bernhard, 2004) described a methodology to set alarm thresholds that takes into account variance between aircraft and various aircraft state parameters (e.g. operating conditions). The work assumed that the underlying data for estimating thresholds have approximately Normal distribution. (Bechhoefer & Bernhard, 2005) further demonstrated that thresholds based on Gaussian statistic could yield greater false alarms than anticipated, and discussed using non-Gaussian distribution, such as Rayleigh distribution for analysis of shaft components. Using a linear transformation to whiten different vibration data types or condition indicators, (Bechhoefer, He, & Dempsey, 2011) presented a method to set a threshold of gear health, also known as health indicator, based on probability of false alarm. The algorithm to define health indicator as a function of condition indicators was developed using three statistical models, namely order statistic, sum of condition indicators, and normalized energy. The models were developed with the assumption that the condition indicators follow Gaussian distribution or Rayleigh distribution.

In the aforementioned work, a lot of investigations were done to determine the most appropriate underlying distribution of the vibration data before a threshold is set. It is often necessary to fit several distributions to the data available, and to choose the most appropriate one based on a goodness-of-fit test, such as in (Jablonski et al., 2013). Rather than trying to fit various distribution functions, it could be more practical to choose a distribution function that can fit a family of distributions, such as Pearson family of distributions and Johnson family of distributions. Thus there are no needs to fit various distribution functions or to compare different thresholds set based on different distributions. This paper focuses on using Johnson family distribution as a unified approach to model a wide variety of distribution functions that describe various vibration data in wind turbine condition monitoring applications. Thus automatic threshold setting could be performed in a more practical manner. Although in condition monitoring system the data available are usually sufficient for statistical analysis, however it is not necessarily true for wind turbine applications due to various seasons or wind conditions that a wind turbine can experience in a year. Ideally at least a whole year is necessary to collect data in order to reflect the true underlying distribution. However it is clearly impractical to collect a year data before condition monitoring system is applied with the correct thresholds. This paper also explores the effects of having limited data available (e.g. a few days, a few weeks, or a few months) in thresholds setting and the possible false alarms generated.

2. JOHNSON FAMILY DISTRIBUTION

Johnson distribution is a family function that can fit different distribution shapes. It is not necessary to test different distributions that will give the best fit to a set of sample data because Johnson family distribution has the flexibility to fit data with a large range of different distribution shapes. A brief description of the Johnson distribution function is provided here.

Fitting data with Johnson distribution involves transforming a continuous random variable x,whose distribution is unknown, into a standard Normal (z) with mean 0 and variance 1 according to one of the four normalizing translations proposed by (Johnson, 1949). The general form of the translation is

$$z = \gamma + \delta f\left(\frac{x-\xi}{\lambda}\right) \tag{1}$$

where $z \sim N(0,1)$, γ and δ are shape parameters, λ is a scale parameter , and ξ is a location parameter. The translation functions that map different distributions to the standard Normal distribution in the Johnson distribution function are

as follows:

$$f(y) = \begin{cases} \ln(y) & \text{for lognormal family}(S_L), \\ \ln\left[y + \sqrt{y^2+1}\right] & \text{for unbounded family}(S_U), \\ \ln\left[\frac{y}{1-y}\right] & \text{for bounded family}(S_B), \\ y & \text{for normal family}(S_N), \end{cases} \tag{2}$$

where $y = \frac{x-\xi}{\lambda}$. If equation 1 is an exact normalizing translation of x to a standard normal random variable, the cumulative density function (CDF) of x is given by

$$F(x) = \Phi(z) \text{ for all } x \in \mathcal{H}, \tag{3}$$

where $\Phi(z)$ denotes CDF of standard Normal distribution, and the space $\mathcal{H}$ of x is

$$\mathcal{H} = \begin{cases} [\xi, +\infty) & \text{for lognormal family}(S_L), \\ (-\infty, +\infty) & \text{for unbounded family}(S_U), \\ [\xi, \xi+\lambda] & \text{for bounded family}(S_B), \\ (-\infty, +\infty) & \text{for normal family}(S_N). \end{cases} \tag{4}$$

The probability density function (PDF) of x is then given by

$$p(x) = \frac{\delta}{\lambda\sqrt{2\pi}} f'(y) \exp\{-\frac{1}{2}[\gamma + \delta f(y)]^2\}, \tag{5}$$

where $f'(y) = \dfrac{df}{dy}$. For more information one can refer to (DeBrota, Dittus, Swain, Roberts, & Wilson, 1989).

There are four methods to estimate Johnson parameters (γ, δ, ξ, λ) as described by (DeBrota et al., 1988), namely: moment matching, percentile matching, least squares, and minimum L_p norm estimation. The reader can refer to (DeBrota et al., 1988) for detailed description of each method. In this work, the moment matching method is used with implementation based on (Hill, Hill, & Holder, 1976) due to its simplicity in Scilab (Enterprises, 2012).

The moment matching method involves determining the family distribution first by the location of skewness, β_1 and kurtosis, β_2 in Figure 1. This figure represents the original identification chart published by (Johnson, 1949), with positive goes downward in the y-axis (β_2). The number of parameters to be estimated is then determined by solving a system of non-linear equations between the sample moments and the corresponding moments of the fitted distribution. A brief procedure of the method can be described as follows:

1. Calculate the moments of x : m_2, m_3 and m_4.
2. Calculate the skewness and kurtosis of x : $\beta_1 \equiv m_3^2/m_2^3$ and $\beta_2 \equiv m_4/m_2^2$.
3. Use the chart in Figure 1 to determine the family or transformation function used.

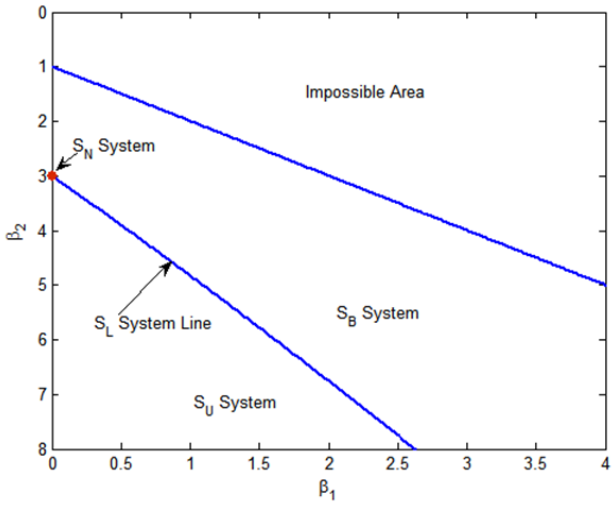


Figure 1. Johnson distribution family identification chart.

3. THRESHOLD SETTING

An alarm threshold can be set based on a predetermined probability of false alarm (pf). This value is essentially a design parameter that can be changed to suit the condition monitoring needs. In this work, the predetermined probability of false alarm is set at 10^{-4}. Thus knowing the underlying probability distribution of the data, it is the same as finding the 99.99 percentile of the distribution or finding the inverse CDF, see equation 6. The inverse CDF of Johnson distribution in this work is computed using Scilab CASCI library, see (Enterprises, 2012).

$$\text{threshold } = F^{-1}(1 - pf). \tag{6}$$

Setting an alarm threshold involves collecting vibration data over a period of time. Depending on how the data are collected, some preprocessing may be needed, such as outliers removal. Next, a probability distribution function is fitted to the data collected and its parameters are estimated. Based on the estimated parameters, a threshold is set following equation 6. Figure 2 illustrates the steps to determine an alarm threshold.

4. DATA COLLECTION FROM A WIND TURBINE

Data used in this study were taken from Generator Non Drive End of a 3 MW turbine. For a typical generator bearing monitoring performed by Brüel & Kjær Vibro (B&K Vibro), there could be up to or more than 10 different vibration data or measurements generated from a sensor. For simplicity of this study, only ISO RMS and High Frequency Band Pass (HFBP) data are used for analysis. HFBP is usually used as early indicator of potential bearing related problems, and ISO RMS

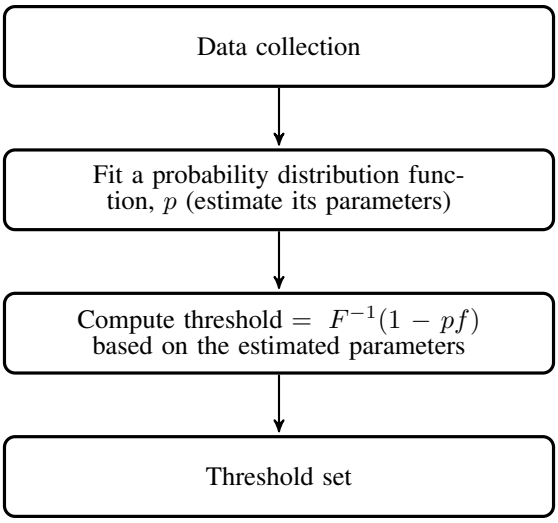


Figure 2. Block diagram of threshold setting.

is usually used as general indicator of faults developing into a later stage. These two measurements or indicators can reflect the general conditions of generator bearings across all turbine types. For more specific problems, such as looseness or imbalance, other measurements or indicators are needed.

ISO RMS and HFBP are computed in the time domain (computing the root mean squared of the signal) after applying the appropriate filter settings. The sample length is set so that it captures approximately 10 revolutions of the generator rotation. The vibration is sampled at 25600 per second.

The data were collected for approximately two months while the turbine was running during its normal operating conditions and producing power at least above 100 kW. No known mechanical faults existed during the data collection period. The data were collected by the data acquisition unit on the turbine and sent every 5 minutes to a remote surveillance center. Data collection interval could actually vary in the real or commercial condition monitoring systems. It often depends on the choice of monitoring strategy of the machine.

As described by (Marhadi & Hilmisson, 2013), since a wind turbine operates over a wide range of speeds and loads, it is important to set thresholds within more or less the same operating condition. Thus changes in measured vibration levels are indeed due to developing faults, and not due to changing operating conditions. Typical B&K Vibro monitoring strategy for wind turbines is to divide the operating conditions of a wind turbine into 5 different operating power classes (OPC) based on the power produced by the wind turbine. For a 3 MW turbine, the power classes are as follow: 100 - 700 kW (Class 1), 700 - 1300 kW (Class 2), 1300 - 2000 kW (Class 3), 2000 - 2700 kW (Class 4), and 2700 - 3200 kW (Class 5). Thus each measurement is classified based on in which operating condition it is taken. No data are recorded when the turbine operates below 100 kW or above 3200 kW.

Figure 3 and 4 present the distributions of ISO RMS and HFBP taken over a period of approximately two months in two power classes. Through out the paper only data from the first two power classes are presented for better clarity and organization. Johnson, Normal, and Weibull distributions are fitted in each type of data for comparison. The figures show that even though the data type is the same (e.g. ISO RMS), however the distribution in different power classes can be significantly different. In this example, the Johnson family type that fits each data type is found to be bounded Johnson distribution (S_B).

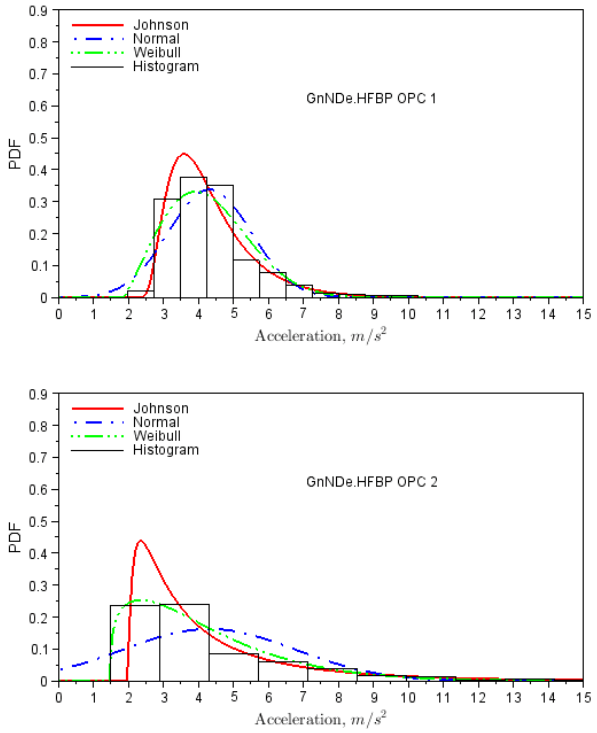


Figure 3. Histogram of HFBP data with different distributions fit in 2 power classes.

The alarm thresholds were then computed following steps described in section 3. In this work, all data are assumed to be valid. Thus no preprocessing (e.g. outliers removal) were done on the collected data. For comparison, table 1 and table 2 present the thresholds of HFBP and ISO RMS calculated based on Johnson, Normal, and Weibull distributions.

5. THRESHOLD CALCULATION BASED ON LIMITED DATA

Ideally, the vibration data collected to set alarm thresholds should reflect all normal operating conditions (without any mechanical faults and the turbine has gone through all possible weather and seasonal conditions) in order to set the thresh-

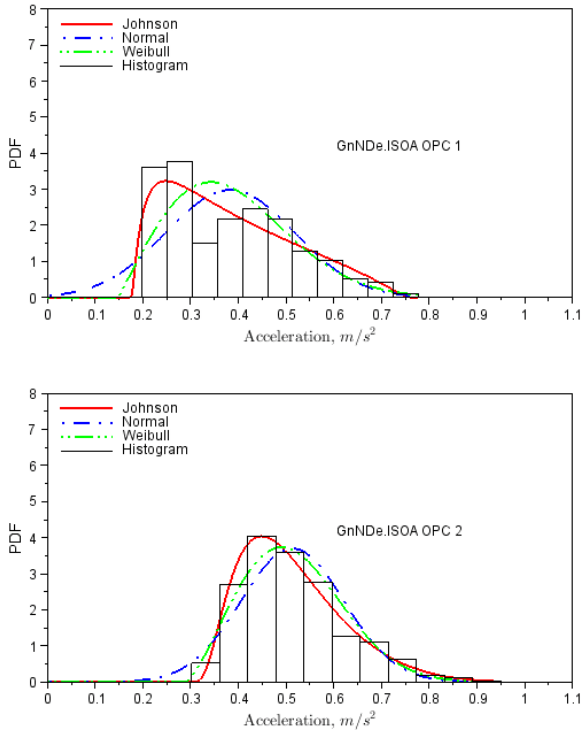


Figure 4. Histogram of ISO RMS data with different distributions fit in 2 power classes.

Table 1. HFBP thresholds at 2 OPCs (m/s^2).

Underlying Distribution	OPC 1	OPC 2
Johnson	12.79	20.75
Normal	8.55	13.18
Weibull	9.19	16.64

olds effectively. This data collection may take up to a year, and it is clearly impractical. A more practical approach is to collect a month of vibration data (or even less than a month), and set the thresholds based on the collected data.

Realistically, the turbine may not have gone through all normal operating conditions after a month of operation. Within almost two months of data collection with every 5 minutes interval of data recording, the numbers of vibration data in each OPC from the turbine used in this study are as follows: 3067 data in OPC 1, 1960 data in OPC 2, 1673 data in OPC 3, 1595 data in OPC 4, and 1719 data in OPC 5. The underlying question is: do these numbers reflect the operating conditions for the rest of the year? Experience has shown that thresholds can be set based on these data, but adjustments might be necessary after a couple of months. For all practical purposes the number of adjustments needs to be minimum.

To investigate the effects of having limited data (not enough data to capture all operating conditions of a turbine) in setting alarm thresholds, the vibration data collected from each OPC are re-sampled uniformly with the following numbers of samples: 720, 360, 180, and 90. It is assumed that the vibration data collected represent the overall operating conditions of the turbine. Another assumption is made that in a worst case scenario, vibration data from a turbine are collected and sent every hour (e.g. to reduce data collection). With this assumption, the vibration data available in this study represent approximately 3 months of data. Then the numbers of re-samples from these data represent 30 days, 15 days, 7.5 days, and 3.75 days of data. Although the numbers of samples look statistically sound, in reality, they may reflect only short periods of the turbine operational time (order of days).

Table 2. ISO RMS thresholds at 2 OPCs (m/s^2).

Underlying Distribution	OPC 1	OPC 2
Johnson	0.77	0.99
Normal	0.89	0.93
Weibull	1.11	0.97

Table 3. HFBP false alarm rates (%) at 2 OPCs when thresholds are set based on the whole data.

Underlying Distribution	OPC 1	OPC 2
Johnson	0.00	0.00
Normal	0.65	1.17
Weibull	0.46	0.31

First, the false alarm rates of the whole data were computed when the thresholds set based on the whole data were used. The results are presented in tables 3 and 4. Thresholds based on Johnson and Weibull distributions generally result in the lowest false alarm rates. However, there are some thresholds that result in false alarm rates that are not within the specified probability of false alarm. Thresholds set based on Normal distribution are more likely to have higher false alarm rate. This shows the difficulty in fitting the most appropriate distribution to the data. For example, the type of Johnson family fitted to the data is bounded (S_B) in all power classes for both HFBP and ISO RMS since the data determine this family to be the most suitable. Having Johnson (S_B) distribution can result in lower thresholds. One can choose to strictly fit Johnson unbounded distribution (S_U) regardless what the data indicate the most appropriate family is, such as in the work done by (Marhadi, Venkataraman, & Pai, 2012). However, having the data determine the most appropriate family and possibly having Johnson S_B distribution as the most appropriate one can prevent the threshold set too high. Thus having a more conservative estimate of the threshold.

Tables 5 and 6 present the false alarm rates when the thresholds set based on limited data are used or checked against the whole data available. As the number of data used to compute thresholds decreases, the false alarm rates can either increase

Table 4. ISO RMS false alarm rates (%) at 2 OPCs when thresholds are set based on the whole data.

Underlying Distribution	OPC 1	OPC 2
Johnson	0.29	0.00
Normal	0.00	0.10
Weibull	0.00	0.00

or decrease. This indicates that the data available are crucial for thresholds setting. Smaller false alarm rates can be achieved if the sampled data are more representative of the actual distribution. Figures 5 and 6 give visual representations of how the distributions of sampled data could actually be different from the whole population.

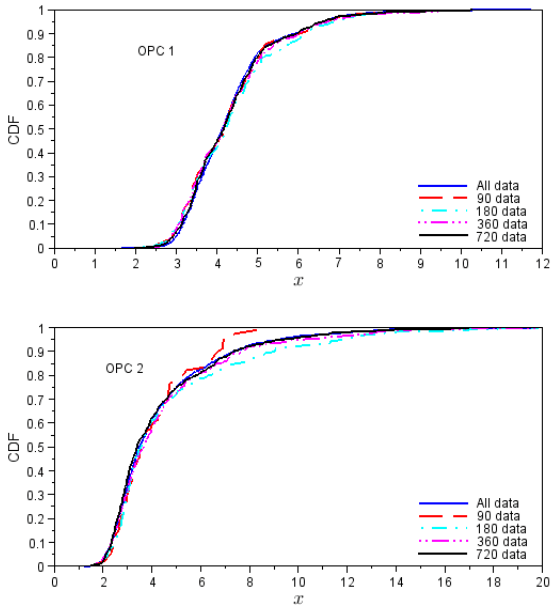


Figure 5. Emperical CDF of HFBP from various sampled data in 2 power classes.

To give some visual representations of the data and how false alarms could occur, figures 7 and 8 show the vibration data over a time period and the thresholds set based on Johnson distribution with different number of data. The figures also show exponential averages of the collected data over time (see Eq. (7)), which can be done to reduce fluctuation in the data and to provide smoother trending. In this study, $\alpha = 0.01$ and $\bar{x}_1 = x_1$.

$$\bar{x}_t = \alpha x_t + (1 - \alpha) x_{t-1} \tag{7}$$

Alarming can be done on the averaged data over time. As stated earlier, the averaged data are smoother and provide a clearer picture when a mechanical fault develops, e.g. by increasing vibration level over time. The false alarm rates are zero in all cases (e.g. different number of samples to set thresholds) when the averaged data are checked against the

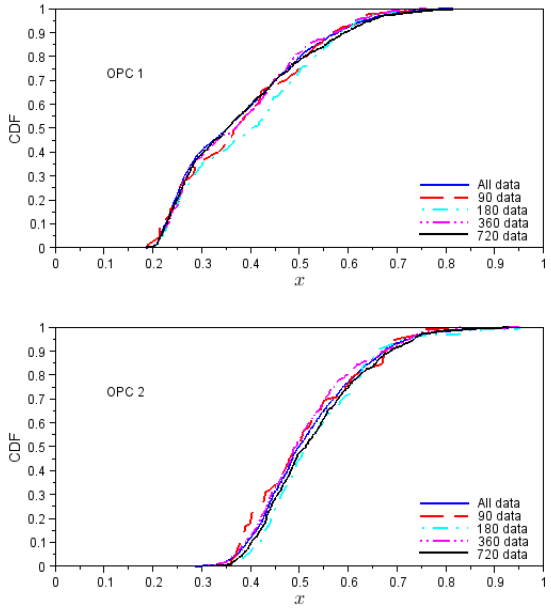


Figure 6. Emperical CDF of ISO RMS from various sampled data in 2 power classes.

Table 5. HFBP false alarm rates (%) at 2 OPCs when thresholds are set based on different number of data.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.00	0.00
	Normal	0.65	0.97
	Weibull	0.36	0.00
360	Johnson	0.00	0.00
	Normal	0.59	0.61
	Weibull	0.13	0.00
180	Johnson	0.00	0.00
	Normal	0.46	0.31
	Weibull	0.20	0.00
90	Johnson	0.00	5.41
	Normal	0.65	3.60
	Weibull	0.13	1.53

computed thresholds. Trending the averaged data also ensures that the machine condition is indeed entering an abnormal condition when the trend crosses a threshold.

Using the averaged data to set thresholds can be done, and will result in thresholds closer to the trend data, which provides quicker response to a change of mechanical condition. However, false alarm rate could be potentially higher, especially when only limited amount of data are available to set the thresholds as illustrated in figures 9 and 10. In these examples, only 720 data are available to set the thresholds, which represent 30 days data collection with every one hour data being sampled, and they are averaged. The false alarm rate in these examples can be as high as 59%. This situation can occur if during the first 30 days of data collection, the frequency of collecting data is not enough to capture many

Table 6. ISO RMS false alarm rates (%) at 2 OPCs when thresholds are set based on different number of data.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.00	0.10
	Normal	0.00	0.10
	Weibull	0.00	0.00
360	Johnson	0.95	0.00
	Normal	0.00	0.20
	Weibull	0.00	0.00
180	Johnson	0.15	0.00
	Normal	0.00	0.10
	Weibull	0.00	0.00
90	Johnson	0.72	1.28
	Normal	0.00	0.10
	Weibull	0.00	0.00

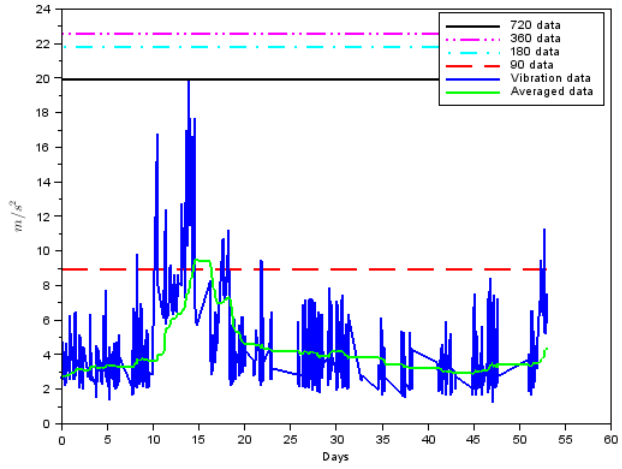


Figure 7. HFBP data over time with thresholds based on fitting Johnson distribution at different number of data in power class 2.

high vibration occurrences. Since the data are averaged, the trend becomes sensitive to high values that are not previously recorded. Thus it is generally more appropriate to use the raw data (without averaging) to set thresholds.

6. QUANTIFYING UNCERTAINTY IN LIMITED DATA

The previous sections have shown that in wind turbine applications, the number of available data can be statistically large, but not necessarily represent the actual distribution of the data or all operating conditions of a turbine. Having limited amount of data generally leads into uncertainty in choosing the appropriate probability distribution to fit the data. Moreover, even if the correct probability distribution is known, having limited amount of data that do not represent the actual population can results in wrong estimates of the distribution parameters. Thus the thresholds set based on these data could be either too low or too high (not optimum).

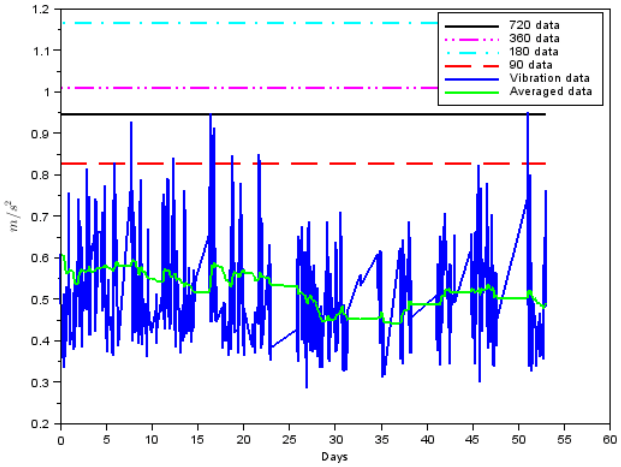


Figure 8. ISO RMS data over time with thresholds based on fitting Johnson distribution at different number of data in power class 2.

It is beneficial to quantify the uncertainty of thresholds (the confidence bounds) set based on limited data. This can be done by first quantifying the uncertainty of the statistical distribution parameters. Different methods are available, both analytically (e.g. maximum likelihood estimate) or based on re-sampling techniques (e.g. bootstrap) and Bayesian estimate. (Marhadi et al., 2012) have described that there have been no analytical methods to estimate uncertainties (confidence bounds) of Johnson distribution fitted to some data. To estimate the uncertainties of the thresholds set based on Johnson distribution (and other distributions in this work), a re-sampling technique (bootstrap) is used. Bootstrap method has relatively simple implementation in comparison to other methods, e.g. Bayesian inference. Although the implementation is simple, bootstrap method is known to have some limitations as described by (Chernick, 1999), such as problems with estimating extreme values and variance of a distribution that has a very large/infinite variance. For comparison and to overcome some of the limitations of bootstrap method, a Bayesian inference procedure is used to estimate the distribution of Johnson function parameters and the resulting bounds of the thresholds.

6.1. Bootstrap Method

Bootstrap technique re-samples the sampled data of 720, 360, 180, and 90 with replacements, and obtains new sets of 720, 360, 180, and 90 data. After each sampling, the distribution parameters are estimated using the selected samples, and the thresholds are calculated based on the estimated parameters of the distributions. Due to sampling with replacement, some samples are repeated in the new selected set. Bootstrap sam-

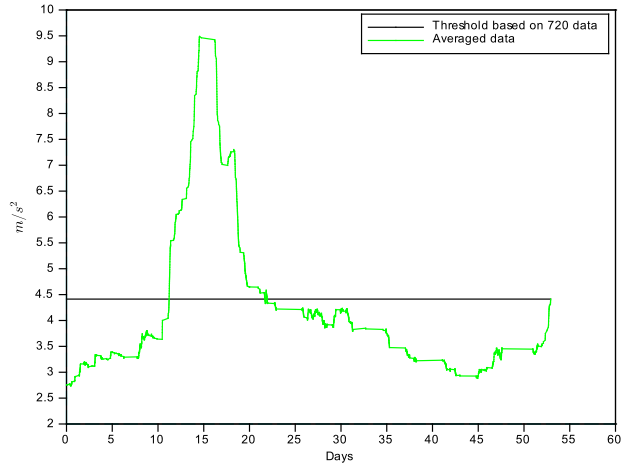


Figure 9. Averaged HFBP data over time with threshold based on fitting Johnson distribution with 720 averaged data in power class 2.

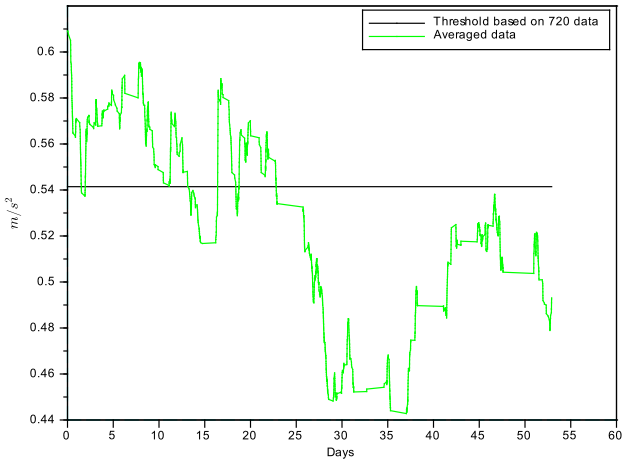


Figure 10. Averaged ISO RMS data over time with threshold based on fitting Johnson distribution with 720 averaged data in power class 2.

pling is applied 1000 times, and the statistical parameters estimated are computed for each sample set in 1000 bootstrap repetition.

For estimating Johnson distribution parameters, in each selection set the appropriate Johnson family distribution (S_L, S_B, S_U, or S_N) is determined using moment values of the data in the selection set. The results of the bootstrap techniques are the 2.5 and 97.5 percentiles of the thresholds set based on each distribution studied. They provide lower and upper bounds of the thresholds with 95% confidence. This information provides flexibility for an engineer to choose the thresholds within the lower and upper bounds.

The false alarm rates are then computed again as the lower and upper bound thresholds are used on the whole data available to simulate a real situation when only limited amount of data available to set thresholds. The results are presented in tables 7 to 10. As one may expect, the lower bound thresholds result in higher false alarm rates and the upper bound ones result in lower rates. Generally the upper thresholds set based on both Johnson and Weibull distributions result in low false alarm rate. The main concern is always whether the thresholds have been set optimally by choosing the most appropriate distribution describing the data. Since the underlying distribution of data collected is not always known beforehand, fitting Johnson distribution can be a general or middle ground solution.

Figures 11 and 12 show the lower and upper bounds (2.5 and 97.5 percentiles) of the thresholds based on Johnson distribution from bootstrapping the 90, 180, 360, and 720 data. They are represented as error bars. Some thresholds determined from limited data are very closed to the thresholds determined from the whole data (e.g. HFBP thresholds in OPC 2 from 180 and 360 data). Some of them are higher or even lower than the thresholds determined from the whole data, but the upper and lower bounds enclose the thresholds from the whole data (e.g. ISO RMS threshold in OPC 2 from 180 data). If the upper bounds are used where they are higher than thresholds set based on the whole data, there is again a concern whether these thresholds are too high or not.

Table 7. HFBP false alarm rates (%) at 2 OPCs when upper thresholds from bootstrap are used.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.00	0.00
	Normal	0.52	0.66
	Weibull	0.13	0.00
360	Johnson	0.00	0.00
	Normal	0.29	0.31
	Weibull	0.00	0.00
180	Johnson	0.00	0.00
	Normal	0.26	0.00
	Weibull	0.00	0.00
90	Johnson	0.00	3.47
	Normal	0.26	2.81
	Weibull	0.00	0.26

6.2. Bayesian Inference of Johnson Distribution

Bayesian procedure is employed to overcome some limitations of the bootstrap method and to address the concern that the upper bound thresholds from bootstrap could be too high or not optimal. The inference of Johnson distribution parameters follows the procedure outlined by (Marhadi et al., 2012). Only Bayesian inference of Johnson distribution parameters

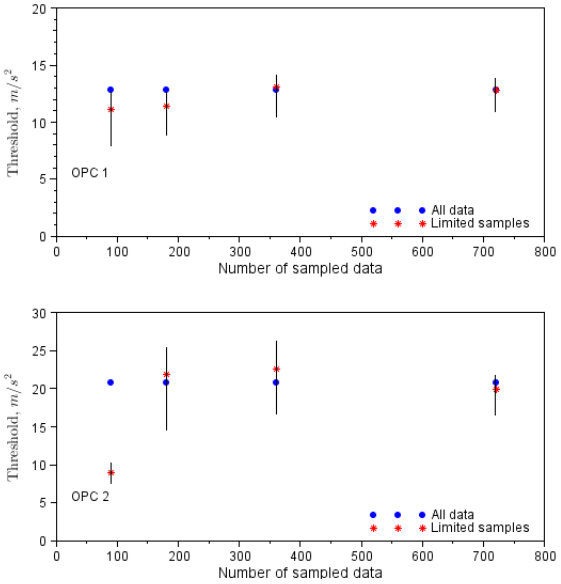


Figure 11. Confidence bounds of HFBP thresholds from bootstrapping various sampled data in 2 power classes. Thresholds are based on Johnson distribution.

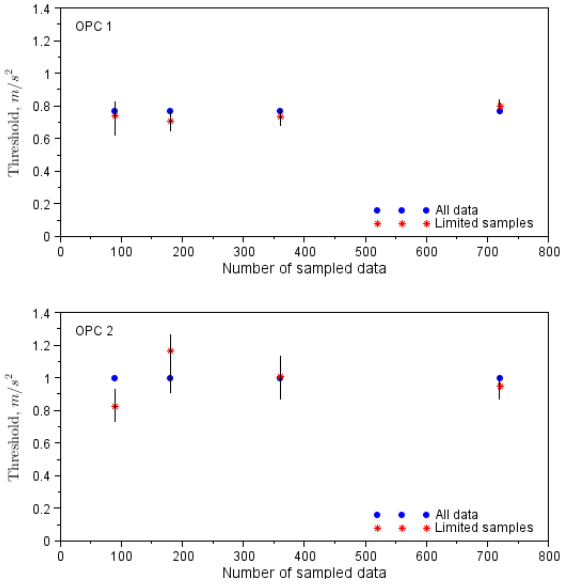


Figure 12. Confidence bounds of ISO RMS thresholds from bootstrapping various sampled data in 2 power classes. Thresholds are based on Johnson distribution.

Table 8. HFBP false alarm rates (%) at 2 OPCs when lower thresholds from bootstrap are used.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.00	0.31
	Normal	0.85	1.58
	Weibull	0.59	0.26
360	Johnson	0.13	0.31
	Normal	0.85	1.53
	Weibull	0.36	0.00
180	Johnson	0.65	0.61
	Normal	0.82	0.87
	Weibull	0.36	0.00
90	Johnson	1.17	9.03
	Normal	1.43	5.56
	Weibull	0.65	3.11

Table 9. ISO RMS false alarm rates (%) at 2 OPCs when upper thresholds from bootstrap are used.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.00	0.00
	Normal	0.00	0.00
	Weibull	0.00	0.00
360	Johnson	0.13	0.00
	Normal	0.00	0.00
	Weibull	0.00	0.00
180	Johnson	0.26	0.00
	Normal	0.00	0.00
	Weibull	0.00	0.00
90	Johnson	0.00	0.10
	Normal	0.00	0.00
	Weibull	0.00	0.00

are considered because this is the focus of the paper, and unlike the other distributions (e.g. Normal and Weibull) there has not been many work on Bayesian inference of Johnson distribution parameters.

Bayesian inference is a statistical method that allows using observation data ($\mathbf{x}$) to infer the unknown parameters ($\boldsymbol{\theta}$) of a distribution that may describe the data. The unknown parameters are represented as PDF. Bayes theorem allows to relate the condition probability distribution of the observed data ($\mathbf{x}$) given the distribution parameters ($\boldsymbol{\theta}$), $p(\mathbf{x}|\boldsymbol{\theta})$ to the condition probability of the parameter ($\boldsymbol{\theta}$) given the observation data ($\mathbf{x}$), $p(\boldsymbol{\theta}|\mathbf{x})$ as shown in equation 8,

$$p(\boldsymbol{\theta}|\mathbf{x}) \propto l(\boldsymbol{\theta}|\mathbf{x})p(\boldsymbol{\theta}), \tag{8}$$

where $p(\boldsymbol{\theta}|\mathbf{x})$ is the posterior PDF of $\boldsymbol{\theta}$ given $\mathbf{x}$, $l(\boldsymbol{\theta}|\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$ is the likelihood of data $\mathbf{x}$ given $\boldsymbol{\theta}$, and $p(\boldsymbol{\theta})$ is known as the prior distributions of $\boldsymbol{\theta}$. The prior here reflects prior knowledge of $\boldsymbol{\theta}$ before any data are considered.

The likelihood is the same as the PDF chosen to fit the data. For Johnson distribution it is equation 5. The prior is usually subjective. The posterior distribution is then obtained by multiplying the prior and all the likelihood functions according to the number of observed data (n) as

$$p(\boldsymbol{\theta}|\mathbf{x}) \propto l(\boldsymbol{\theta}|x_1)l(\boldsymbol{\theta}|x_2)\dots l(\boldsymbol{\theta}|x_n)p(\boldsymbol{\theta}). \tag{9}$$

Sampling the joint distribution function (posterior distribution) in equation 9 is often difficult and required using a Markov Chain Monte Carlo (MCMC) method. In (Marhadi et al., 2012), they used a Metropolis method to sample the posterior

Table 10. ISO RMS false alarm rates (%) at 2 OPCs when lower thresholds from bootstrap are used.

Number of data	Underlying Distribution	OPC 1	OPC 2
720	Johnson	0.59	0.56
	Normal	0.00	0.26
	Weibull	0.00	0.00
360	Johnson	2.38	0.56
	Normal	0.00	0.51
	Weibull	0.00	0.20
180	Johnson	4.47	0.31
	Normal	0.00	0.46
	Weibull	0.00	0.20
90	Johnson	5.80	4.59
	Normal	0.00	0.61
	Weibull	0.00	0.41

distribution. They also chose to use non-informative prior or flat prior, with an infinite interval. They reported that sampling the four parameters of Johnson distribution simultaneously could cause the Metropolis method fail to converge. It is more likely to achieve convergence by inferring only two parameters, namely γ and δ assuming the estimates for location and scale parameters (ξ and λ) are more accurate to obtain.

Following findings in (Marhadi et al., 2012), only γ and δ are inferred in this work. Based on the sampled data, Bayesian inference of Johnson S_B, S_L, S_N or S_U distribution can be performed. It is determined based on the moments of the data using moment matching method as described in section 2. Bayesian inference is performed with a random walk Metropolis method with 4000 burn-in iterations period and 2000 samples from the posterior distribution. The scale parameters (variance) of the proposal distribution/density (a bivariate Normal distribution with zero covariance) are adjusted so that acceptance rate between 30% to 50% can be achieved. For more details description of the Metropolis method, one can refer to (MacKay, 2003). It is found that even when only γ and δ are inferred in this work, convergence of the Metropolis method can be difficult to achieve when flat prior is used. Thus Normal priors for γ and δ are investigated. Again, prior is often subjective and could be subject to more detailed investigation in future work.

It is assumed that γ and δ are distributed according Normal distribution. The means are assumed to be equal to the first estimates of γ and δ of the sampled data. The variance is difficult to estimate. However, after some trials and errors, it is found that standard deviations of 0.5 of the means (first estimates of γ and δ) could result in satisfactory convergence. Figure 13 shows the output of 2000 samples for γ and δ from the Metropolis method after 4000 burn-in iteration with 90 data from ISO-RMS at OPC 2. The running average plotted in the figure (green line) shows convergence of the method. The initial estimates of the parameters for these 90 data are as follows: $\gamma = 0.644$, $\delta = 0.807$, $\xi = 0.339$, $\lambda = 0.499$, and the Johnson distribution family is S_B or bounded. Samples from the Metropolis method have means of $\gamma = 0.624$ and $\delta = 0.806$. In this work, all of the limited sampled data fall into the family of S_B or bounded Johnson distribution. Thus in this work Bayesian inference is done mainly with Johnson S_B family distribution.

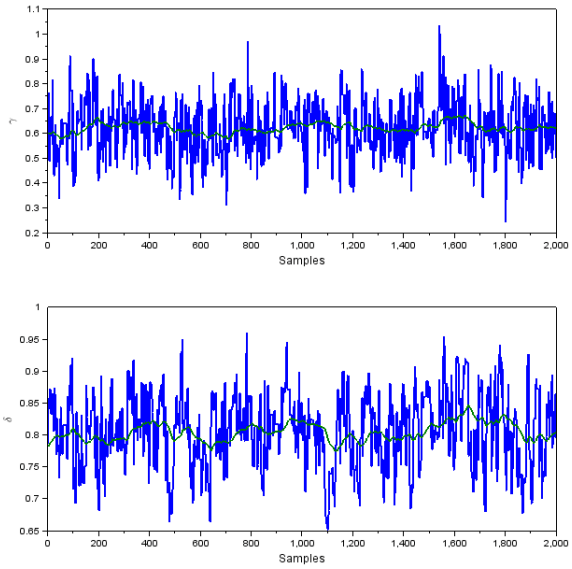


Figure 13. 2000 samples of γ and δ from Metropolis method after 4000 burn-in iteration with 90 data from ISO-RMS at OPC 2. ξ and λ are kept constant at the initial estimates.

The 2000 samples of parameters estimated from Bayesian inference are then used to determine thresholds based on Johnson distribution. The 2.5 and 97.5 percentiles of the thresholds are determined as in the case when bootstrap is used to provide lower and upper bounds. Figures 14 and 15 show the lower and upper bounds of the thresholds based on Bayesian inference of 90, 180, 360, and 720 data. In comparison to results from bootstrap, the bounds for HFBP are generally larger, with the lower bounds are generally much lower, which could result in much higher false alarm rates if they are used. Only in OPC 1 where HFBP thresholds from 360 and 720 data have much higher upper bounds than the bounds from bootstrap. These results could be due to the choice of prior, which is subject to further study. On the contrary, the bounds for ISO RMS are generally much tighter than bounds from bootstrap. These results are encouraging to prevent setting thresholds too high. For completeness, the false alarm rates are computed again as the lower and upper bound thresholds are used on the whole data available. The results are presented in tables 11 to 14.

Using Bayesian inference to quantify uncertainties in setting alarm thresholds is actually attractive when large quantity of historical data are available because the method facilitates

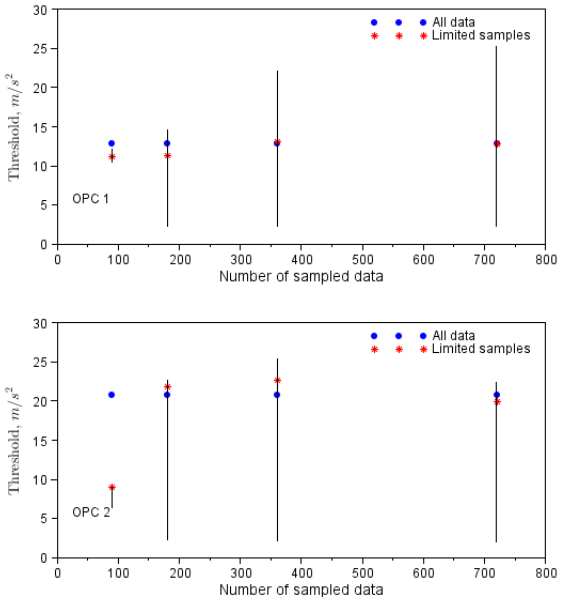


Figure 14. Confidence bounds of HFBP thresholds from Bayesian inference of various sampled data in 2 power classes. Thresholds are based on Johnson distribution.

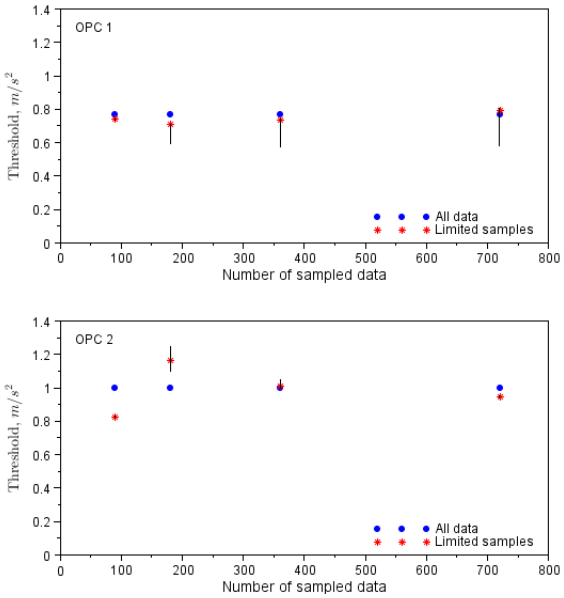


Figure 15. Confidence bounds of ISO RMS thresholds from Bayesian inference of various sampled data in 2 power classes. Thresholds are based on Johnson distribution.

Table 11. HFBP false alarm rates (%) at 2 OPCs when upper thresholds from Bayesian inference are used. Underlying distribution is Johnson.

Number of data	OPC 1	OPC 2
720	0.00	0.00
360	0.00	0.00
180	0.00	0.00
90	0.00	5.41

Table 12. HFBP false alarm rates (%) at 2 OPCs when lower thresholds from Bayesian inference are used. Underlying distribution is Johnson.

Number of data	OPC 1	OPC 2
720	99.7	97.5
360	99.7	96.0
180	99.7	93.0
90	0.13	15.4

learning. However there are still some challenges that need to be solved before it can be used in real industrial applications, such as having a faster/efficient method to sample the posterior distribution. In case of using an MCMC method, there is not yet a well established method to determine how many burn-in iterations are needed that guarantees convergence. Convergence could potentially be achieved after a long burn-in period that requires long computational time. In regards to using Johnson distribution, proper selection of the priors still needs further investigation so that sampling the posterior distribution is computationally efficient, and the whole 4 parameters could possibly be inferred.

In the actual wind turbine condition monitoring at B&K Vibro, an alarm is not always generated when a measurement crosses a threshold in any power classes. A more complex system is implemented to prevent false alarms, see for example the work by (Marhadi & Hilmisson, 2013). This paper simply presents a general framework to set alarm thresholds automatically using Johnson distribution, and how the uncertainties in setting the thresholds can be quantified when only limited data are available. The method could be useful not only in wind turbine applications, but also in other machineries.

7. CONCLUSION

A method to set alarm thresholds automatically based on fitting Johnson distribution to vibration data has been presented. Using Johnson distribution eliminates the need to test various distributions that could fit the collected data most appropriately. Thus it can prevent choosing incorrect distribution that may result in setting sub-optimal thresholds. Results in this study show that low false alarm rate can be achieved by utilizing Johnson distribution. The implementation is simple and straightforward, which should also be applicable in machineries other than wind turbines.

The problem of having limited data in wind turbines that may not represent the whole or most operating conditions of a turbine has been investigated based on bootstrap method and Bayesian inference. Lower and upper bounds of alarm thresholds are obtained using both methods, and the false alarm

Table 13. ISO RMS false alarm rates (%) at 2 OPCs when upper thresholds from Bayesian inference are used. Underlying distribution is Johnson.

Number of data	OPC 1	OPC 2
720	0.00	0.00
360	0.59	0.00
180	1.50	0.00
90	0.59	1.07

Table 14. ISO RMS false alarm rates (%) at 2 OPCs when lower thresholds from Bayesian inference are used. Underlying distribution is Johnson.

Number of data	OPC 1	OPC 2
720	10.2	0.10
360	10.6	0.00
180	8.80	0.00
90	1.04	1.32

rates are investigated when these thresholds are used. These could provide information where to set the thresholds effectively. Bootstrap is generally simple to implement, while Bayesian inference has slightly more complicated implementation. However, initial results in this study suggest that Bayesian inference could potentially prevent from setting the thresholds too high once the challenges of its implementation can be overcome.

Future work may include investigation of the effectiveness of the method when it is actually implemented to a wide number of turbines to catch real mechanical faults. Comparison with other methods or the more established ones could be made in this way, and the effectiveness of each method can be validated. Future work may also include finding the most effective method to estimate Johnson distribution parameters other than the moment matching method used in this study.

REFERENCES

Andersson, C., Gutt, S., & Hastings, M. (2007). Cost-effective monitoring solution using external surveillance centre. In *The second world congress on engineering asset management and the forth international conference on condition monitoring.* Harrogate, UK.

Bechhoefer, E., & Bernhard, A. P. (2004). Setting hums condition indicator thresholds by modeling aircraft and torque band variance. In *Aerospace conference, 2004. proceedings. 2004 ieee* (Vol. 6, pp. 3590–3595).

Bechhoefer, E., & Bernhard, A. P. (2005). Use of non-gaussian distribution for analysis of shaft components. In *Aerospace conference, 2006 ieee* (pp. 9–pp).

Bechhoefer, E., He, D., & Dempsey, P. (2011). Gear health threshold setting based on a probability of false alarm. In *Annual conference of the prognostics and health management society, september 25–20, montreal, canada.*

Cempel, C. (1987). Simple condition forecasting techniques in vibroacoustical diagnostics. *Mechanical Systems and Signal Processing*, *1*(1), 75–82.

Cempel, C. (1990). Limit value in the practice of machine vibration diagnostics. *Mechanical systems and signal processing*, *4*(6), 483–493.

Chernick, M. R. (1999). *Bootstrap methods, a practitioners guide*. John Wiley & Sons.

Crabtree, C. (2011). *Condition monitoring techniques for wind turbines*. Unpublished doctoral dissertation, Durham University.

DeBrota, D. J., Dittus, R. S., Swain, J. J., Roberts, S. D., & Wilson, J. R. (1989). Modeling input processes with johnson distributions. In *Proceedings of the 21st conference on winter simulation* (pp. 308–318).

DeBrota, D. J., Roberts, S. D., Swain, J. J., Dittus, R. S., Wilson, J. R., & Venkatraman, S. (1988). Input modeling with the johnson system of distributions. In *Proceedings of the 20th conference on winter simulation* (pp. 165–179).

Enterprises, S. (2012). Scilab: Free and open source software for numerical computation [Computer software manual]. Orsay, France. Retrieved from `http://www.scilab.org`

Hill, I., Hill, R., & Holder, R. (1976). Algorithm as 99: Fitting johnson curves by moments. *Applied Statistics*, 180–189.

Jablonski, A., Barszcz, T., Bielecka, M., & Breuhaus, P. (2013). Modeling of probability distribution functions for automatic threshold calculation in condition monitoring systems. *Measurement*, *46*(1), 727–738.

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, *16*, 149–176.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.

Marhadi, K., & Hilmisson, R. (2013, 11–13 June). Simple and effective technique for early detection of rolling element bearing fault: A case study in wind turbine application. In *International congress of condition monitoring and diagnostic engineering management (comadem).* Helsinki, Finland.

Marhadi, K., Venkataraman, S., & Pai, S. S. (2012). Quantifying uncertainty in statistical distribution of small sample data using bayesian inference of unbounded johnson distribution. *International Journal of Reliability and Safety*, *6*(4), 311–337.

Tavner, P. (2012). *Offshore wind turbines*. The Institution of Engineering and Technology.

BIOGRAPHIES

Kun S. Marhadi is an engineer in the Remote Monitoring Group at Brüel & Kjær Vibro. He joined Brüel & Kjær Vibro in 2012. Previously, he worked as a postdoctoral fellow in the Department of Mathematics at the Technical University of Denmark (DTU). He received his PhD in computational science in 2010 from San Diego State University and Claremont Graduate University. He has M.S. and B.S. in aerospace engineering from Texas A&M University. His expertise is in structural vibration and analyses, probabilistic methods, and design optimization.

Georgios Alexandros Skrimpas was born in Athens, Greece in 1986. He received the Diploma in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2009 and the M. Sc. in wind energy from the Technical University of Denmark (DTU) in 2012. He joined Brüel and Kjær Vibro in 2012 and since 2013 he is pursuing the Industrial Ph.D. degree at the Centre of Electric Power and Energy at DTU in cooperation with Brüel and Kjær Vibro. His research interests are diagnosis and prognosis of electrical and mechanical faults in wind turbines.

Temporal Causal Diagrams for Diagnosing Failures in Cyber-Physical Systems

Nagabhushan Mahadevan[1], Abhishek Dubey[1], Gabor Karsai[1], Anurag Srivastava[2], and Chen-Ching Liu[2]

[1] *Institute for Software-Integrated Systems, Vanderbilt University, Nashville, TN 37212, USA*
nag,dabhishe,gabor@isis.vanderbilt.edu

[2] *The School Of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99163, USA*
asrivast,liu@eecs.wsu.edu

ABSTRACT

Resilient and reliable operation of cyber physical systems of societal importance such as Smart Electric Grids is one of the top national priorities. Due to their critical nature, these systems are equipped with fast-acting, local protection mechanisms. However, commonly misguided protection actions together with system dynamics can lead to un-intentional cascading effects. This paper describes the ongoing work using Temporal Causal Diagrams (TCD), a refinement of the Timed Failure Propagation Graphs (TFPG), to diagnose problems associated with the power transmission lines protected by a combination of relays and breakers.

The TCD models represent the faults and their propagation as TFPG, the nominal and faulty behavior of components (including local, discrete controllers and protection devices) as Timed Discrete Event Systems (TDES), and capture the cumulative and cascading effects of these interactions. The TCD diagnosis engine includes an extended TFPG-like reasoner which in addition to observing the alarms and mode changes (as the TFPG), monitors the event traces (that correspond to the behavioral aspects of the model) to generate hypotheses that consistently explain all the observations. In this paper, we show the results of applying the TCD to a segment of a power transmission system that is protected by distance relays and breakers.

1. INTRODUCTION

Cyber-Physical Systems (CPS) such as the Smart Electric Grids are going through transformational reform powered by federal funding and in line with the stated national energy security mission goals (Garrity, 2008). These systems work in dynamic environments resulting from varying load, changing operational requirements and conditions, physical component degradation, and software failures. To reach the required level of resiliency and reliability, efficient online management of CPS is necessary to operate safely within specified parameters, even in the presence of faults (Ilic et al., 2005). One aspect of online management is fault identification, diagnostics, prognostication, and mitigation. Inability to automatically and timely diagnose and pinpoint the source(s) of failures combined with the potential side-effects of automated protection actions lead to impending fault cascades, which can be avoided (Zhang, Ilic, & Tonguz, 2011; Tholomier, Richards, & Apostolov, 2007). Recent blackouts and hurricane Sandy in 2012 demonstrated the grid vulnerability and reasons to look at existing defense mechanism more closely.

Fast acting localized protection mechanisms are used arrest the propagation of failure effects. Electrical protection systems include detection devices such as fast-acting relays that are designed to detect abnormal changes in physical properties (current, voltage, impedance) and actuation devices such as breakers that can be triggered to open the circuit in electrical networks. To observe, track, and possibly diagnose these systems, it is important to consider the discrete and continuous dynamics of the physical system, the protection systems and their interactions both in the nominal and faulty modes of operations. During nominal (fault-free) operation, both physical and protection systems should operate nominally to provide the desired functionality. If a fault appears in the physical system, the nominal protection system is expected to detect the failure effect and isolate the faulty part of the system. In some cases, the nominal protection system is assisted by a set of algorithms to restore the system functionality to its original configuration once the physical fault disappears (due to a temporary fault or after repair).

Operators have to consider the possibilities of misoperations of protection systems. Distance relays have been known to incorrectly initiate tripping due to an apparent impedance that

Nagabhushan Mahadevan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

fall into the Zone settings of line relays caused by heavy load and depressed voltage conditions (Pourbeik, Kundur, & Taylor, 2006). In fact, an investigation by North Electric Reliability Corporation (NERC) demonstrated that nearly all major system events, excluding those caused by severe weather, have had relay or automatic control misoperations (almost 2,000 in one year) contributing to worsening the impact of failure propagation (North American Electric Reliability Corporation, 2012). Protection malfunction and its correlation with major blackouts require a careful rethinking of its system-wide effects (Zhang et al., 2011; Pourbeik et al., 2006).

This paper describes Temporal Causal Diagrams (TCD), a refinement of the Timed Failure Propagation Graphs (TFPG) (Abdelwahed, Karsai, Mahadevan, & Ofsthun, 2009), to diagnose failures of physical systems that are instrumented with multiple local fast acting protection devices and controllers to isolate the faults. The TCD is a discrete abstraction that captures the causal and temporal relationships between failure modes (causes) and discrepancies (effects) in a system, thereby modeling the failure cascades taking into account propagation constraints imposed by operating modes, protection elements, and timing delays. Faults and their propagation are captured using TFPG models, the nominal and faulty operations of the components (controllers, protection devices etc.) are captured as Timed Discrete Event Systems (TDES). We also present a diagnosis reasoner that extends the TFPG diagnosis algorithm considering both the alarms and mode changes (as reported by the physical system), as well as the various event traces corresponding to the behavioral aspects of the mode. The uniqueness of the approach is that it does not involve complex real-time computations involving high-fidelity models, but performs reasoning using efficient graph algorithms based on the observation of various anomalies and events in the system. When fine-grained results are needed and computing resources and time are available, the diagnostic hypotheses can be refined with the help of the physics-based diagnostics.

The paper is organized as follows. The next section (Section 2) deals with the related research. Section 3 that describes the temporal causal diagrams. Section 4 documents the results of applying the solution to various fault scenarios in a power transmission system and Section 5 concludes the paper with a discussion of the future work. Notations used and an overview of Timed Failure Propagation Graphs (TFPG) are described in appendices.

2. RELATED RESEARCH

Fault diagnostics has been recognized as a critical task in electric grid operations (Coster, Myrzik, Kruimer, & Kling, 2011). A classic but excellent summary of power system fault diagnostics is provided in (Sekine, Akimoto, Kunugi, Fukui, & Fukui, 2002), including Bayesian approaches (Mengshoel et al., 2010; Yongli, Limin, & Jinling, 2006), rule-based reasoning (Meléndez et al., 2004; Lee et al., 2004), expert systems (Talukdar, Cardozo, & Perry, 2007; Yang, Okamoto, Yokoyama, & Sekine, 1992), fuzzy-logic methods (W. Chen, Liu, & Tsai, 2000; Sun, Qin, & Song, 2004), Genetic Algorithm, search based techniques (Lin, Ke, Li, Weng, & Han, 2010), artificial neural network (Guo et al., 2010; Zhou, 1993), and Petri Nets by abstracting the power system as a discrete event system (Sun et al., 2004) (Ren, Mi, Zhao, & Yang, 2005). Problems similar to large electric system operations also occur in smaller systems such as Electric Ship (Bastos, Zhang, Srivastava, & Schulz, 2007) and Spacecraft (Poll et al., 2007; Daigle et al., 2010).

A pioneering paper (Fukui & Kawakami, 1986) reports a rule-based or logic-based system for location of line faults based on real time information acquired at the control center of a power system. (Sekine et al., 2002) compiled a comprehensive survey of the fault diagnostics systems developed using various knowledge-based system techniques. Model-based approaches based on logic behaviors of the protection devices are identified as valuable tools for fault analysis. The on-line alarm analyzer reported in (Miao, Sforna, & Liu, 1996) incorporates the cause-effect principles of protective devices into logic-based proof-oriented algorithms for the analysis of malfunctions. Cause-effect models are used for fault diagnostics of substations in (W.-H. Chen, Liu, & Tsai, 2000). Upon field-testing with real world data it was found that the proofs are difficult when uncertainties cannot be resolved. The proof algorithm in (Miao et al., 1996) had to be generalized in order to evaluate the credibility of potentially large number of hypotheses (W.-H. Chen et al., 2000).

The approach described in this paper differs from existing practice where fault analysis and mitigation relies on a logic-based approach that relies on hard thresholds and local information assisted by manual system level analysis. The causal model presented in this paper is based on the timed failure propagation graph (TFPG) introduced in (Misra, 1994; Misra, Sztipanovits, & Carnes, 1994), which is conceptually related to the temporal causal network approach presented in (Console & Torasso, 1991; Padalkar, Sztipanovits, Karsai, Miyasaka, & Okuda, 1991; Karsai, Sztipanovits, Padalkar, & Biegl, 1992; Mosterman & Biswas, 1999). The TFPG model was extended in (Abdelwahed, Karsai, & Biswas, 2004) to include mode dependency constraints on the propagation links, which can then be used to handle failure scenarios in hybrid and switching systems.

We have extended this work to be able to take local mitigation in a subsystem, especially in case of malfunction of protection devices results in a larger fault cascade, leading to a blackout into consideration. This is primarily done by considering the discrete behavior of the protection devices and using it in the diagnosis. The problem of fault diagnosis in discrete

event systems has been extensively studied. According to (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1996), the fault diagnosis problem can be described in terms of a description of a plant's behavior in the form of a finite automaton Any behavior of the plant can be represented as a run of this automaton, i.e. a sequence of events. These events can be either *observable* or *unobservable*. If the fault event is observable then the diagnosis problem is trivial. However, usually one or more unobservable events correspond to the occurrence of a fault that may occur in the plant operation. The objective is to find a diagnoser that can detect the occurrence of a fault event within a bounded number of steps from the occurrence. However, we need to consider the possibility of timed failure propagation and faults in the controllers as well as plant.

Our approach can improve the effectiveness of isolating failures in large-scale systems such as Smart Electric Grids, by identifying impending failure propagations and determining the time to critical failure, which increases the system reliability and reduce the losses accrued due to power failures.

3. TEMPORAL CAUSAL DIAGRAMS

A Temporal Causal Diagram is a behavior augmented temporal failure propagation graph model. The TCD model of a component can describe the fault propagation and/ or the behavior. The failure propagation is described in terms of Timed Failure Propagation Graphs (TFPG)[1]. The component behavior under nominal and faulty conditions is captured through Timed Discrete Event Systems (TDES). A TDES is characterized as follows:

- Q: The set of discrete states of the component
- F: The set of failure modes internal to the component. As always, failures modes are not directly observable.
- D: The set of discrepancies, i.e. potentially observable anomalies, if any, associated with the component behavior. The discrepancy can be detected, or triggered by the component, or affect the component behavior.
- Σ: The set of events that correspond to controller commands, actuation, external mode commands, detection of the physical state of component, discrepancy detection or other internal events. The detection of a discrepancy, d, is written as d$\uparrow$, while d$\downarrow$ relates to the remission of a discrepancy.
- A mode map, $\mathbb{M} : Q \rightarrow 2^M$ captures the effect of a state in Q on the TFPG-mode in M. Thus, the system being in a discrete state affects the current modes of the TFPG, which in turn affects the propagation link.
- δ is the transition map. The transitions are written as $[Guard]Event(delay)/Actions$. The $Guard$ condition can represent the presence of a local fault $f \in F$, written as $in(f)$ and absence of it, written as $!in(f)$. Note that for brevity, unless specifically required we will use the shorthand f and $!f$ in the guard conditions. $Actions$ result in production of events that can be communicated to the rest of the system, and/or change the mode of the system. $delay$, if present declares that the transition will occur after the timeout. The rising edge of the event is described by appending the uparrow $\uparrow$ to event. The falling edge of the event is shown using the downarrow $\downarrow$.

[1] See appendix A for an overview on TFPG

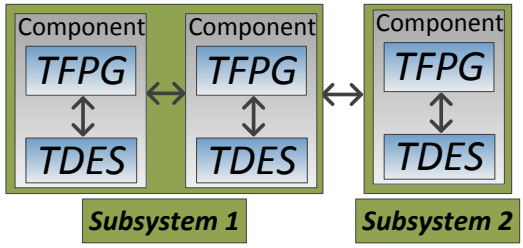


Figure 1. A TCD model of a system consists of interacting subsystems containing components, where each component consists of an interacting TFPG and TDES model.

Figure 1 provides an overview of the TCD model of a system. The TCD model is hierarchical where a system model is composed of subsystem models which in themselves are composed of component models. The component model includes TFPG and/ or TDES models. The TCD model captures the interactions between the TFPG and TDES models both within the component, as well as across component boundaries. The interactions between the TFPG and TDES models are captured implicitly through the state changes in the common modeling elements in the two models - failure modes, discrepancies, and modes. The behavioral model can be designed to consume and react to the updates of these common elements in the form of events (appearance, disappearance, change) and conditions (presence, absence). Likewise, the behavioral model can be designed to update these common elements that can be consumed by the failure propagation model. The cascading failure propagation effects across component boundaries is captured explicitly (as in TFPG) through failure propagation links between the discrepancy elements in each component. Interactions between the behavior models are based on the event generation and consumption paradigm. A TDES component can consume events corresponding to commands, detection, and mode changes generated by one or more component TDES models. It can also generate similar events to be consumed by other component TDES models.

Example 1 *An example illustrative TCD model is shown in the Figure 2. The failure modes (F1, F2, F3) are shown as rectangular blocks and the discrepancies (D1, D2, D3, D4, D5, D6) as circular elements. The fault propagation across the TFPG model is captured by the edges between the faults and the discrepancies. The markers (M1, M2) on the edges*

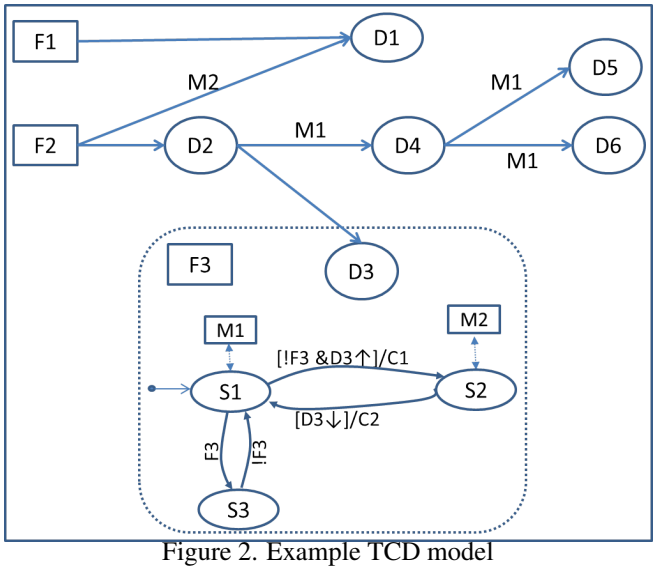


Figure 2. Example TCD model

capture the mode in which the fault could propagate via the edge. Edges that do not carry any mode marker are always enabled implying the faults can propagate in any mode (M1 or M2) across these edges.

The dotted-box captures a behavioral TDES model of a protection element. It captures three operational states: S1, S2, and S3. S1 is the initial state which maps to system mode M1. The protection element transitions from state S1 to S2 when it detects the presence of a discrepancy D3 and the fault F3 is not present (guard condition: $!F3\&D3\uparrow$) and issues a command (event) C1. The state transition results in a mode change to M2. This nominal operation of the protection element arrests the propagation of the failure effect due to fault F2, thereby preventing the anomalies related to discrepancies D4, D5, D6 from triggering in the system. However, it could happen that the anomaly related to discrepancy D1 is observed in the system.

Also, the TDES model shows that when the protection element detects the absence of the discrepancy D3 (transition: $D3\downarrow$)), it issues a command C2 (event) and transitions back to the state S1 (and restores the system mode back to M1). If the fault F2 were to reappear and trigger discrepancy D3, the protection element would react again to arrest the fault propagation.

Fault F3 captures an internal fault in the protection element with regards to detecting the presence of D3. The TDES model captures this as the protection element transitioning into state S3. When the fault F3 disappears, the protection element is automatically restored to the nominal state S1. However, when in S3 the protection element cannot react to the presence of the discrepancy D3 and hence cannot arrest the fault propagation leading to the triggering of anomalies related to discrepancies D4, D5, and D6.

3.1. Event Propagation Paths from the Behavioral Model

The TDES models in TCD are used to generate event propagation paths. An event propagation path is generated for each transition and state when the transition parameters (trigger, guard, action) or state parameters (entry/ exit/ during actions) include event variables that belong to any of the following categories: failure mode, discrepancy, or observable events: detection, command, and actuation. When these variables are present in the event and/ or guard condition, they are treated as (causal) source nodes of the event propagation path. When they are present in the transition actions and state actions (entry/during), they are treated as the destination (effect) nodes. The modes appear as source (destination) nodes, if they are mapped to the source (destination) state in the TDES model. Additional nodes in the event propagation path include composition nodes (*AND* and *OR*) that relate/ combine the cause(s) (source nodes) and effect(s) (destination nodes), as well as *NOT* nodes that are used to mark absence or disappearance of faults (i.e. failure modes). Multiple event propagation paths can be chained together by tracing the state-transition model in the TDES and ignoring the internal, unobservable states and events.

Example 2 *Event propagation paths for the protection element TDES model in Figure 2 are*
(a) $M1, !F3, D3\uparrow \rightarrow C1, M2$, *(b)* $M2, D3\downarrow \rightarrow C2, M1$, *and (c)* $M1, F3 \rightarrow \varnothing(NoObs)$.

3.2. Reasoning using TCD

The TCD reasoning algorithm relies on the fault propagation model (TFPG) and the event propagation models (generated from the TDES) to hypothesize the possible causes for the anomalies and event traces observed in the system. The algorithm tries to explain the observations in terms of a consistency relationship between the states of the nodes and edges in the fault propagation and event propagation model.

The TCD reasoning algorithm considers the *physical*, *observed* and *hypothetical* states of the nodes and edges in the fault propagation and event propagation model. A *physical* state corresponds to the current state of the set (V) of all the nodes and edges.. At any time t, the physical state of the nodes and edges is given by a map $AS_t : V \rightarrow \{\texttt{ON}, \texttt{OFF}\} \times \mathbb{R}$. An ON state for a fault node indicates that the failure is present, otherwise it is set to OFF. For a discrepancy node, an ON state indicates that the failure (effect) has reached this node, otherwise it is set to OFF. An ON state for a failure propagation edge indicates that the edge can carry the failure (effect) from the parent to the child node, otherwise it is set to OFF. For the non-failure nodes from the event propagation models, an ON state indicates that the associated *event-variable* or *mode-variable* is set to the state represented by that node, otherwise the state is OFF.

The *observed state* at time t is defined as a map $S_t : V \rightarrow$

Algorithm 1 TCD Reasoner Update

1: **INPUTS**: t, HS_{t-1}, O_t.
2: $HS_t = UpdateHypo(t, HS_{t-1})$
3: **if** $O_t \neq \varnothing$ **then**
4: $HS_t^{'} = HS_t$
5: $HS_t = \varnothing$
6: **for all** $H \in HS_t^{'}$ **do**
7: **if** Consis(H,O_t) **then**
8: $HS_t \leftarrow HS_t \cup \{H\}$
9: **end if**
10: **end for**
11: **if** $HS_t \neq \varnothing$ **then**
12: **for all** $H \in HS_t^{'}$ **do**
13: $HS_t \leftarrow HS_t \cup ExplainHypo(H, O_t)$
14: **end for**
15: **end if**
16: **end if**
17: **return** HS_t

$\{\texttt{ON}, \texttt{OFF}\} \times \mathbb{R}$, for all the observable nodes in the fault and event propagation model. The aim of the TCD reasoning process is to find a consistent and plausible explanation of the current system *physical* state based on the *observed* state. Such explanation is given in the form of a valid hypothetical state. A *hypothetical state* is a map that defines the states of the node (and edges) and the interval at which each node(and edges) changes its state. Formally a hypothetical state at time t is a map $H_t^{V'} : V' \rightarrow \{\texttt{ON}, \texttt{OFF}, \texttt{UNKNOWN}\} \times \mathbb{R} \times \mathbb{R}$ where $V' \subseteq V$.

A reasoner hypothesis is an estimate of the current state of all nodes in the system and the time period at which each node changed its state. An estimate of the current state is valid only if it is consistent with the TCD model. State consistency in TCD model is a node-parent relationship that can be extended pairwise to arbitrary subsets of nodes. The TCD reasoner uses the consistency relationships defined in (Abdelwahed et al., 2004; Abdelwahed, Karsai, & Biswas, 2005) (between the TFPG nodes and edges) for all the nodes and edges in the TCD model, i.e. it extends the consistency relationship to the non-fault nodes in the event propagation model as well. At any time, t, during the reasoning process, the TCD reasoner uses the Algorithm 1 to update the hypotheses based on the current set of observations. Algorithm 1 uses extended versions of the concepts and algorithms defined in (Abdelwahed et al., 2004, 2005) to account for event propagation and consistency in event nodes. The additional procedures invoked by the algorithm are briefly described in the appendix A.

Inputs to the TCD Diagnosis Algorithm 1 include the current time, t , the prior hypotheses set, HS_{t-1}, and the current alarm and event observations, O_t. The diagnosis algorithm (1) returns a set hypotheses that can consistently explain the current observed state of the TCD system. The algorithm starts by updating the existing hypotheses (HS_{t-1}) to the current time HS_t (line #2). Then, it identifies the set of hypotheses that can consistently explain the current alarm and event observations (lines #4-#9). In case none of the hypotheses are consistent with the observations, the algorithm generates new hypotheses from each of the old hypothesis to explain the current observations (lines #10 - #16). Across each update, the TCD reasoner keeps a score of the number of consistent, inconsistent, missing, and pending observations for each hypothesis and generates metrics (described later) to identify the best possible explanation, i.e. hypothesis.

Hypotheses Ranking

The quality of the generated hypotheses is measured based on three independent factors: (*a*) *Plausibility* is a measure of the degree to which a given hypothesis group explains the current fault and event signature. (*b*) *Robustness* is a measure of the degree to which a given hypothesis is expected to remain constant. (*c*) *#FM* is a measure of how many failure modes are listed by the hypothesis. The reasoner prefers parsimony principle (minimal number of failure modes) to report results. (*d*) *Failure rate* is a measure of how often a particular failure mode will occur. In case of multiple failures, the failure rates of failure modes are combined assuming independence.

3.3. Reasoner improvements

The improvements and updates in the TCD reasoning process over the TFPG reasoner include: (*a*) Observation evolution, i.e. tolerating the evolution or change in the *observed* state of the nodes. (*b*) Internal mode changes, i.e. accounting for mode changes that are not externally controlled but introduced by the dynamics of the protection systems. The mode change could be unobservable, but inferred based on other observations. (*c*) Fault negation, i.e. accounting for disappearance or absence of one or more faults based on certain observations.

Handling changes in the observations

In case of the TFPG reasoner, the *observed* state of a discrepancy node is either considered latched or intermittent (due to the nature of the fault or problems in the sensor). However in TCD, the dynamics of the protection system might prevent a certain failure propagation and hence result in an apparently consistent change to the *observed* state of an alarm (or discrepancy). It is also possible that the both appearance and disappearance of a fault can be accounted for when the *observed* state of the discrepancy is allowed to change. More importantly, since the protection systems are actively trying to arrest the failure effect propagation and also respond to the disappearance of faults, it is possible that the *observed* state of the non-fault event nodes could be updated over time based on the behavioral model of the protection system. If the events are observable, then the TCD reasoner updates the *hypothetical* states to be consistent with the update *observed* state of the fault and non-fault nodes. In the TCD example shown in Figure 2, it is possible that when the fault F2 happens, the

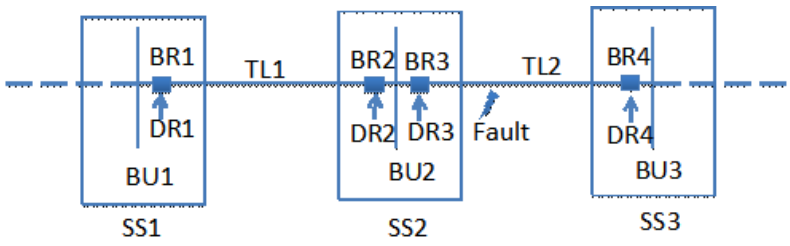


Figure 3. Segment of a Power Transmission System

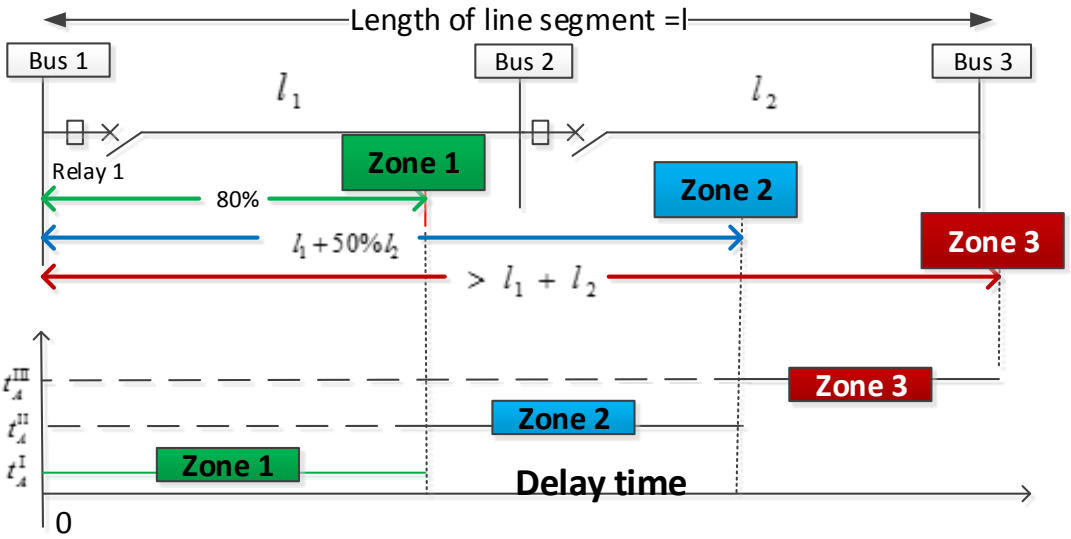


Figure 4. Protection Zone Configuration for Distance Relay. Zone 1 is set to protect 80% of the entire length of the line, and operates immediately (t_A^I) if the fault falls in the zone 1 protection region. Zone 2 is set to protect 100% of the entire line length plus at 50% of the adjacent line, and operates with time delay, t_A^{II}, 15-30 cycles. (0.5s). Zone 3 is set to protect 100% of the entire line length plus at 100% of the adjacent line, and operates with time delay , t_A^{III} (1.5s)

anomalies D4, D5, D6 could have triggered because the system was in mode M1. However, once the protection system completes its operation and the mode is changed to M2, the anomalies related to D4, D5, D6 should not be observable or detectable (based on the model). The TCD reasoner can account for this by changing the *hypothetical* states of these nodes to `UNKNOWN`. Further, later on if the mode is restored to M1 when D3 disappears (!D3)), the reasoner can account for disappearance (or lack of observation) of D2, D4, D5 and D6. This is done by applying the consistency relationship to update the *hypothetical* state of fault F2, discrepancy D2, D4, D5, and D6 to `OFF`.

Mode changes introduced by protection system

The protection and control systems are actively involved in changing the mode of the physical system to arrest the fault propagation. The TCD reasoning algorithm accounts for this by allowing for a *hypothetical* state for each mode. The *hypothetical* state of the mode is updated based on other observations and the consistency relationship between the *hypothetical* states of the mode with other TCD nodes. The reasoning algorithm updates the expected *hypothetical* states of other nodes if the *hypothetical* state of the mode changes. In the TCD example shown in Figure 2, the TCD reasoner updates the *hypothetical* states based on the mode changes introduced by the protection system. In case the mode is changed to M2 upon appearance of the fault F1, the updated *hypothetical* state for D1 can consistently explain any observation of anomaly related to D1. In case, the protection system fault F3 is present, then the lack of any observation (NULL) from the protection system and observations of discrepancy D4, D5, D6 would suggest that the system is still in mode M1 and the protection system has failed to act because of fault, F3.

Fault negation

The TCD reasoning algorithm can generate hypotheses that state that one or more faults are not present in the system. This is possible if the TDES model (and hence the event propagation model) includes specific conditions that state certain events can happen only if the fault is not present. The event propagation model accounts for the negated fault, and updates the hypothesis appropriately if the concerned events are ob-

Table 1. Fault Propagation: The faults in the transmission lines are categorized based on the segment where they occur along the length(L) of the line (from left to right) - F_20:$[0, 0.2L)$, F_50: $[0.2L, 0.5L)$, F_80: $[0.5L, 0.8L)$, F_100: $[0.8L, 1.0L)$, where L is the length of the transmission line. The row in the table should be read as described for the first row: A fault F_20 in transmission line TL1 will lead to a zone 1 fault (d_z1) in DR1, a zone 2 fault (d_z2) in DR2 and a zone 3 fault (d_z3) in DR3.

Source Node (Transmission Line. Failure Mode)	Destination Node (Relay.zone)	Mode
TL1.F_20	DR1.d_z1, DR2.d_z2, DR4.d_z3	M_Close
TL1. F_50	DR1._dz1, DR2.d_z1, DR4.d_z3	M_Close
TL1.F_80	DR1.d_z1, DR2.d_z1, DR4.d_z2	M_Close
TL1. F_100	DR1.d_z2, DR2.d_z1, DR4.d_z2	M_Close
TL2. F_20	DR1.d_z2, DR3.d_z1, DR4.d_z2	M_Close
TL2.F_50	DR1.d_z2, DR3.d_z1, DR4.d_z1	M_Close
TL2. F_80	DR1.d_z3, DR3.d_z1, DR4.d_z1	M_Close
TL2.F_100	DR1.d_z3, DR3.d_z2, DR4.d_z1	M_Close

served. In the TCD example shown in Figure 2, the triggering of command C1 by the protection system indicates among other things the absence of fault, F3. Also, the triggering of command C2, indicates the disappearance of D3 (!$D3$) and hence the negation or disappearance of the fault F2.

4. EXAMPLE

The example system considered in this paper (Figure 3) is a segment of a power transmission system. Power system components such as buses, lines, transformers, are protected by relays and breakers. When a fault occurs, relays and breakers are designed to isolate the fault according to a pre-determined protection scheme. Additionally, the system includes back-up relays to account for any problems in the primary relays and breakers. The system in Figure 3 is part of a network and includes three substations(SS1, SS2, and SS3) and two transmission lines (TL1,TL2). Transmission line TL1 carries power between buses BU1 and BU2 while transmission line TL2 is between buses BU2 and BU3. Each transmission line is protected with a distance relay and breaker at its two ends.

The distance relays estimate impedance using the voltage and current measurement at the relay measurement point. The estimated impedance is compared with the reach point impedance. If the estimated impedance is less than the reach point impedance, it is assumed that a fault exists on the line between the relay and the reach point. The fault-zone (zone1, zone2, zone3) is determined based on the estimated impedance. Figure 4 shows the region corresponding to each protection zone relative to Relay DR1 and the relative time-scales for the relay operation in each zone. A distance relay has to perform the dual task of primary and back up protection depending on the fault zone. For faults in zone1 (80% of the entire length of the transmission line (L1)), it serves as the primary protection and acts fast without any intentional time delay (($t_A{}^1$ = 5 to 6 cycles). For faults in zone2 (up to 50% of the adjacent line) and zone3 (up to 100% of the adjacent line), the relay serves as a back-up and reacts with some time delay allowing for the primary relay to operate. In Zone2, the time delay ($t_A{}^2$)) is approximately 15-30 cycles (0.5 sec), while in Zone3 it acts with a delay ($t_A{}^3$)) of about 1.5 sec. Additionally, to account for temporary faults in the transmission lines, the relays include a fast and delayed auto-reclosure function, wherein they check for the fault after 2 sec (fast reclosure) and after 2-3 minutes (delayed reclosure). In case the faults persist, the relay disconnects the circuit permanently until it is remotely commanded to reset.

Each substation has a remote terminal unit (RTU) as part of the SCADA system to send the breaker status and other measurements to control center's Energy Management System (EMS). Some of the details recorded by the Sequence Event Recorder (SER) at each substation include: (*a*) Zone information and start protection time (in case of zone 1) (*b*) Tripping command sent by relay to breaker (*c*) Breaker status: opened or closed (*d*) Phase discordance problem: when breaker tried to open three phases but did not succeed for all three phases (*e*) Reclosure command issued by the relay to reclose breaker (*f*) Reclosure blocked command issued by relay to reset breaker to open after failed reclosure.

4.1. TCD model

The TCD model of the system in Figure 3 includes a) fault propagation model for transmission line faults, b) the breaker behavioral model and (c) the distance relay behavioral model.

Fault Propagation Model: Table 1 captures the propagation of the faults in the transmission lines (TL1, TL2) to the discrepancies in distance relays (DR1, DR2, DR3, DR4). The faults in the transmission lines are categorized based on the segment where they occur along the length(L) of the line (from left to right) - F_20:$[0, 0.2L)$, F_50: $[0.2L, 0.5L)$, F_80: $[0.5L, 0.8L)$, F_100: $[0.8L, 1.0L)$, where L is the length of the transmission line. Discrepancies correspond to the zone with respect to the relay - d_z1: zone1, d_z2: zone2, d_z3: zone3. All failure propagations are active in mode M_Close when the circuit is closed.

Breaker Behavioral Model: The breaker behavioral model (table 2) includes states Open, Close, and partially open. The Open state maps to the system mode M_Open, states Close

Table 2. Transitions in a breaker's behavior model. The model includes states Open, Close and partially open (P_Open). Close is the initial state. Rows 1-2 capture the nominal operation to close and open the breaker. Rows 3-11 deal with faulty operation - rows 3,4:stuck close fault, rows 5-6:stuck open fault, rows 7-11: partially open fault.

#	Src. State	Dst. State	Trigger	Guard	Action
1	Open	Close	C_Close	!F_st_open &!F_part	St_Close
2	Close	Open	C_Open	!F_st_close &!F_part	St_Open
3	Open	Close	F_st_close	none	none
4	Close	Close	C_Open	F_st_close	St_Close
5	Close	Open	F_st_Open	none	none
6	Open	Open	C_Close	F_st_open	St_Open
7	Open	P_Open	F_part	none	none
8	P_Open	Open	!F_part	none	none
9	Close	P_Open	C_Open	F_part	St_Open
10	P_Open	P_Open	C_Open	F_part	St_Open
11	P_Open	Close	C_Close	none	St_Close

Table 3. Transition Information for Distance Relay's behavioral model. Rows 1-7 deal with the anomaly detection in state Det (rows 1-3: Zone1, rows 4,5: Zone2, rows 6,7: Zone3). Rows 8,9 deal with wait (until timeout) operation in Wait state based on the wait time T_w set for different operations - fast-reclosure(TFR), delayed-reclosure (TDR), backup in zone2 (Tw2) and zone3 (Tw3). Row 10-12 deal with system mode conditions for anomaly detection (transition to state Det). Rows 13-16 handle resets. Rows 17-21 deal with anomaly detection fault (F_de).

#	Src State	Dst State	Trigger	Guard	Action
1	Det	Wait	d_z1↑	n=0	Z1, C_Open, n=1, Tw=TFR
2	Det	Wait	d_z1↑	n=1	C_Open, FRBLK, n=2, Tw=TDR
3	Det	BLK	d_z1↑	n=2	C_Open, DRBLK
4	Det	Wait	d_z2↑	n=0	n=3, Tw=Tz2
5	Det	BLK	d_z2↑	n=3	C_Open
6	Det	Wait	d_z3↑	n=0	n=4, Tw=Tz3
7	Det	BLK	d_z3↑	n=4	C_Open
8	Wait	Ch_Det	$Timeout(T_w)$	n <= 2	C_Close
9	Wait	Ch_Det	$Timeout(T_w)$	n > 2	none
10	Ch_det	Det	none	M_Close& !F_de	none
11	Ch_det	No_Det	none	M_Open	none
12	No_Det	Det	none	M_Close	none
13	No_Det	Reset	C_Reset	none	none
14	BLK	Reset	C_Reset	none	C_Close
15	Det	Reset	d_z1↓ &d_z2↓ &d_z3↓ & n>0	none	none
16	Reset	Ch_det	none	none	n=0
17	Ch_det	Det_Err	F_de	none	none
18	Det_Err	Ch_Det	!F_de	none	none
19	Det	Det_Err	F_de	none	none

and P_Open (partially open) map to the mode M_Close. The breaker receives commands from its distance relay to open (C_Open) and close (C_Close). After executing the command, it reports the physical state of the breaker as St_open (for open) and St_close (close). The behavioral model includes breaker faults related to being stuck open (F_st_open), stuck close (F_st_close) and partially open (F_part). Table 2 shows the operation of the breaker in terms of the transitions between the states based on the events (commands) and fault conditions. Rows 1-2 capture the nominal operation to close and open the breaker when it receives the appropriate command. While rows 3-4 capture the breaker behavior when it is stuck close, rows 5-6 deal with a breaker with a stuck open fault. Rows 7-11 deal with a partially open breaker (which leads to phase discordance problems in the system).

Event propagation paths related to the transitions listed in Table 2 capture the pre (source) and post (destination) conditions and observations to help analyze whether the breaker is operating nominally or is faulty. The generated event propagation paths are as follows:

(*a*) M_Close, C_Open, !F_st_close, !F_part → St_Open, M_Open
(*b*) M_Open, C_Close, !F_st_Open,!F_part → St_Close, M_Close
(*c*) M_Open, C_Close, F_st_Open → St_Open, M_Open
(*d*) M_Close, C_Open, F_st_Close → St_Close, M_Close
(*e*) M_Close, C_Open, F_part → St_Open, M_Close
(*f*) M_Close, C_Close, F_part → St_Close, M_Close

Distance Relay: The behavioral model states include: (a) Det: state when it is actively looking for anomalies and triggering appropriate action upon detection, (b) Wait: when it is waiting for a time-out to expire before taking the next set of actions (c) BLK: when it is blocking and waiting for a reset command as it has taken the necessary action to arrest

Table 4. Scenario 1: Distance Relays - Events and Hypotheses

Time(s)	Comp	Event	Hypotheses
100.02	DR3 DR4 DR1	Z1, C_Open Z2	$H1_{DR3}$=d_z1, M:1/1 $H1_{DR4}$=d_z1, M:1/1 $H1_{DR1}$=d_z2 $H1_{sys}$=TL2.F_20, M:2/3 $H2_{sys}$=TL2.F_50, M:3/3 $H3_{sys}$=TL2.F_80, M:2/3 $H4_{sys}$=TL2.F_100, M:1/3
102.04	DR3, DR4	C_Close	
102.07	DR3, DR4	FRBLK, C_Open	$H2_{sys}$= TL2.F_50, M: 5/5
222.09	DR3, DR4	C_Close	
222.12	DR3, DR4	DRBLK, C_Open	$H2_{sys}$= TL2.F_50, M: 7/7

Table 5. Scenario 1: Breakers - Events & Hypotheses

Time(s)	Comp	Event	Hypotheses
100.03/ 102.08/ 202.13	BR3, BR4	C_Open, St_Open	$H1_{BR3}$=C_Open, M_Open $H1_{BR4}$=C_Open, M_Open
102.05/ 222.10	BR3, BR4	C_Close, St_Close	$H2_{BR3}$=C_Close, M_Close $H2_{BR4}$=C_Close, M_Close

Table 6. Event trace and Hypotheses: Scenario 2

Time (s)	Comp	Event	Hypotheses
100.02	DR3 DR4 DR1	Z1 C_Open Z2	$H1_{DR3}$=d_z1, M:1/1 $H1_{DR4}$=d_z1, M:1/1 $H1_{DR1}$=d_z2 $H1_{sys}$=TL2.F_20, M:2/3 $H2_{sys}$=TL2.F_50, M:3/3 $H3_{sys}$=TL2.F_80, M:2/3 $H4_{sys}$=TL2.F_100, M:1/3
102.07	DR3, DR4	NULL (No Obs)	$H1_{DR3}$=d_z1, M:1/2 $H1_{DR4}$=d_z1, M:1/2 $H2_{DR3}$=d_z1↓,d_z2↓,d_z3↓,M:1/1 $H2_{DR4}$=d_z1↓,d_z2↓,d_z3↓,M:1/1 $H2_{sys}$= TL2.F_50, M: 3/5 $H3_{sys}$= !TL2.F_50, M: 2/2

the fault propagation, (d) Det_Err: when it is unable to detect anomalies because of internal fault (F_de), (e) other miscellaneous states such as Ch_det (where it checks if detection is feasible), No_Det (when no detection is possible), Reset (when it is resetting).

The distance relays detects anomalies pertaining to faults in Zone1 (d_z1), Zone2 (d_z2) and Zone3 (d_z3) of the appropriate transmission line and reports these observations through output-events Z1 (Zone1), Z2 (Zone2) and Z3 (Zone3) respectively. It issues commands to the breaker to open (C_Open) and close (C_Close) and acts upon command to reset (C_reset). It reports unsuccessful fast and delayed re-closure through the output events FRBLK and DRBLK respectively. The faults considered as part of the distance relay include failure to detect the anomalies in transmission line impedance (F_de). While the distance relay states do not map to any system-modes, the system-modes determine if the distance relay is capable of detecting anomalies (mode: M_Close) or not (Mode: M_Open).

Tables 3 describe the transitions for the distance relay's behavioral model. The rows 1-3 deal with the nominal operation when discrepancy related to zone1 fault is detected (row 2: fast re-closure, row 3: delayed re-closure). Rows 4,5 deal with zone2 fault and rows 6,7 with zone3 fault. The wait time (T_w) in the Wait state are set for fast reclosure (TFR), delayed reclosure (TDR), backup wait time in zone2 fault (Tz2) and zone3 fault (Tz3). These wait times (T_w) are used in the $TIMEOUT(T_w)$ operation in rows 8 and 9. Rows 10,11,12 specify the system modes in which the distance relay can detect anomalies i.e. transition to Det state. Rows 13-16 deal with resetting the distance relay. Rows 17-21 deal with presence or disappearance of fault (F_de) related to problems in detecting anomalies.

Event propagation paths related to the transitions listed in Table 3 capture the pre (source) and post (destination) conditions and observations to help analyze whether the distance relay is operating nominally or is faulty. The generated event propagation paths are as follows:
(*a*) M_Close, d_z1↑ → Z1, C_Open (*b*) M_Close, d_z1↑ → FRBLK, C_Open (*c*) M_Close, d_z1↑ → DRBLK, C_Open (*d*) M_Close, d_z2↑ → Z2 (*e*) M_Close, d_z2↑ → C_Open (*f*) M_Close, d_z3↑ → Z3

Table 7. Event trace and Hypotheses: Scenario 3

Time (s)	Comp	Events	Hypotheses
100.02	DR4	Z1,C_Open	$H1_{DR4}$=d_z1, M:1/1
	DR1	Z2	$H1_{DR1}$=d_z2 $H1_{sys}$=TL2.F_50, M:2/2 $H2_{sys}$=TL2.F_80, M:1/2 $H3_{sys}$=TL2.F_100, M:1/2
100.07	DR1	C_Open	$H1_{DR3}$=F_de, M:1/1 $H4_{sys}$= TL2.F_50, DR3.F_de M: 3/3
102.07	DR4	FRBLK, C_Open	$H4_{sys}$= TL2.F_50, DR3.F_de, M: 4/4
222.12	DR4	DRBLK, C_Open	$H4_{sys}$= TL2.F_50, DR3.F_de, M: 5/5

(*g*) M_Close, d_z3↑ → C_Open (*h*) F_de → NULL (No Obs)
(*i*) d_z1↓ & d_z2↓ d_z3↓ → NULL (No Obs)

4.2. Case Study: Fault Scenarios and Diagnosis Results

This section considers a few of fault scenarios in the example power transmission system (Figure 3). The discrete behavioral and fault propagation model described in the Section 4.1 are used to simulate the system both in the nominal and faulty modes. The simulation is performed in Acumen (Taha et al., 2012) with a simulation time-step of 0.01 sec. The observable event-traces are collected and analyzed based on the algorithm 1. The reasoner uses the event propagation paths described in in Section4.1 to reason about the events observed in the breakers (BR1, BR2, BR3, BR4) and distance relays (DR1, DR2, DR3, DR4). The fault propagation model captured in Table1 is used to produce system-wide consistent hypotheses that can explain the observed anomalies and event traces.

In all the scenarios described below, the system is considered to be operating in nominal mode (mode=M_Close) until time t=100sec, when transmission line, TL2 experiences a line-to-ground-short fault, F_50.

Scenario 1: Permanent Fault In Transmission Line

In this scenario, the fault (TL2.F_50) is persistent. The simulator generated event-traces (similar to data from Sequence Event Recorders in real system) are fed to the TCD reasoner. Table 4, presents the events observed from the distance relays (DR1,DR3, DR4) and the hypotheses generated by TCD reasoner. The initial hypotheses point towards a zone1 discrepancy (d_z1) in DR3, DR4 and zone2 discrepancy in (d_z2) in DR1. System level hypotheses, $H2_{sys}$ (fault: TL2.F_50) has the maximum metric (3/3) with three consistent evidences from DR1,DR3,DR4. Moving forward, the observations of failed reclosure - fast (FRBLK) and delayed (DRBLK) - from DR3, DR4 further support $H2_{sys}$ (7/7), suggesting a diagnosis of fault in F_50 in TL2.

The events generated from the breaker and their associated hypotheses are presented in Table 5. The hypotheses suggest nominal operation and capture the mode-change. The multiple time values in each row of column 1 correspond to different times when the same event (& hypotheses) are observed.

Senario 2: Temporary Fault In Transmission Line Here, the fault ($TL2.F_50$) lasts for exactly 1 sec. DR3, DR4 come-up to test the fast re-closure 2 sec after detecting a zone 1 discrepancy (d_z1). Hypotheses $H2_{DR3}, H2_{DR4}$ identify the lack of any observations to be consistent with the event propagation path corresponding to the disappearance of discrepancies (d_z1↓, d_z2 ↓, d_z3↓). Thereafter system hypotheses $H3_{sys}$ suggests with a 100% (2/2) supporting evidences that there is no fault in TL2 (!TL2.F_50)

Scenario 3: Fault In Transmission Line and Relay This is a multi-fault scenario in which a distance relay fault, F_de, prevents DR3 from detecting discrepancies produced by transmission line fault, TL2.F_50. Lack of observations consistent with the predicted hypothetical state of DR3.d_z1 suggest problems with the event propagation path (M_Close, d_z1, !F_de) in DR3. Hypothesis $H1_{DR3}$ in Table 7 explains this observation (or lack of), with fault DR3.F_de. The multi-fault system hypothesis ($H4_{sys}$) best explains the observations.

5. DISCUSSION AND CONCLUSION

We have presented in this paper a new formalism: Temporal Causal Diagrams - with the objective of applying it to diagnose cyber-physical systems that include local fast-acting protection devices. Specifically, we have demonstrated the capability of the TCD model to capture the discrete fault propagation and behavioral model of a segment of a power transmission system protected by distance relays and breakers. Further, the paper presented the potential of the TCD-based reasoner to diagnose faults in the physical system and its protection elements.

As part of our future work, we wish to test and study the scalability of this approach towards a larger power transmission system including a far richer set of protection elements. Further, we wish to consider more realistic event traces from the fault-scenarios including missing, inconsistent, and out-of-sequence alarms and events.

ACKNOWLEDGMENT

This work is funded in part by the National Science Foundation under the award number CNS-1329803. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

NOMENCLATURE

t	arbitrary time instant
A_t	Alarms observed at time t
Ev_t	Events observed at time t
O_t	Observations (Alarms and Events) at time t
H	Hypothesis - a data structure that captures the hypothetical states of all the nodes in the model.
HS_t	Hypotheses set at time t.
$HS_t^{'}$	Temporary variable - hypotheses set.
$\uparrow$	rising edge of an event. Also used to describe the onset of a discrepancy.
$\downarrow$	falling edge of an event. If associated with a discrepancy it describes the event associated with the remission of the discrepancy.

REFERENCES

Abdelwahed, S., Karsai, G., & Biswas, G. (2004). System diagnosis using hybrid failure propagation graphs. In *The 15th international workshop on principles of diagnosis.* Carcassonne, France.

Abdelwahed, S., Karsai, G., & Biswas, G. (2005). A consistency-based robust diagnosis approach for temporal causal systems. In *The 16th international workshop on principles of diagnosis.* Pacific Grove, CA.

Abdelwahed, S., Karsai, G., Mahadevan, N., & Ofsthun, S. (2009). Practical Implementation of Diagnosis Systems Using Timed Failure Propagation Graph Models. *Instrumentation and Measurement, IEEE Transactions on*, *58*(2), 240–247.

Bastos, J. L., Zhang, Y., Srivastava, A. K., & Schulz, N. N. (2007). A design paradigm for integrated protection of shipboard power systems. In *Proceedings of the 2007 summer computer simulation conference* (pp. 3:1–3:10). San Diego, CA, USA: Society for Computer Simulation International.

Chen, W., Liu, C., & Tsai, M. (2000). On-line fault diagnosis of distribution substations using hybrid cause-effect network and fuzzy rule-based method. *Power Delivery, IEEE Transactions on*, *15*(2), 710–717.

Chen, W.-H., Liu, C.-W., & Tsai, M.-S. (2000, Apr). On-line fault diagnosis of distribution substations using hybrid cause-effect network and fuzzy rule-based method. *Power Delivery, IEEE Transactions on*, *15*(2), 710-717. doi: 10.1109/61.853009

Console, L., & Torasso, P. (1991). On the co-operation between abductive and temporal reasoning in medical diagnosis. *Artificial Intelligence in Medicine*, *3*(6), 291-311.

Coster, E., Myrzik, J., Kruimer, B., & Kling, W. (2011, January). Integration issues of distributed generation in distribution grids. *Proceedings of the IEEE*, *99*(1), 28 -39. doi: 10.1109/JPROC.2010.2052776

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., & Poll, S. (2010). A Comprehensive Diagnosis Methodology for Complex Hybrid Systems: A Case Study on Spacecraft Power Distribution Systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *40*(5), 917–931.

Fukui, C., & Kawakami, J. (1986, Oct). An expert system for fault section estimation using information from protective relays and circuit breakers. *Power Delivery, IEEE Transactions on*, *1*(4), 83-90. doi: 10.1109/TPWRD.1986.4308033

Garrity, T. (2008). Getting smart. *Power and Energy Magazine, IEEE*, *6*(2), 38–45.

Guo, W., Wen, F., Ledwich, G., Liao, Z., He, X., & Liang, J. (2010). An Analytic Model for Fault Diagnosis in Power Systems Considering Malfunctions of Protective Relays and Circuit Breakers. *Power Delivery, IEEE Transactions on*, *25*(3), 1393–1401.

Ilic, M., Allen, H., Chapman, W., King, C., Lang, J. H., & Litvinov, E. (2005, Nov). Preventing future blackouts by means of enhanced electric power systems control: From complexity to order. *Proceedings of the IEEE*, *93*(11), 1920-1941. doi: 10.1109/JPROC.2005.857496

Karsai, G., Sztipanovits, J., Padalkar, S., & Biegl, C. (1992). Model based intelligent process control for cogenerator plants. *Journal of Parallel and Distributed Systems*, *15*, 90–103.

Lee, S., Choi, M., Kang, S., Jin, B., Lee, D., Ahn, B., ... Wee, S. (2004). An intelligent and efficient fault location and diagnosis scheme for radial distribution systems. *Power Delivery, IEEE Transactions on*, *19*(2), 524–532.

Lin, X., Ke, S., Li, Z., Weng, H., & Han, X. (2010). A Fault Diagnosis Method of Power Systems Based on Improved Objective Function and Genetic Algorithm-Tabu Search. *Power Delivery, IEEE Transactions on*, *25*(3), 1268–1274.

Meléndez, J., Macaya, D., Colomer, J., Llanos, D., Gervas, P., & Gupta, K. (2004). Symptom based representation for dynamic systems diagnosis. Application to Electrical Power Distribution. In *Proceedings of the eccbr workshops. edited by p. gervas and km gupta. university of madrid, madrid* (pp. 311–327).

Mengshoel, O., Chavira, M., Cascio, K., Poll, S., Darwiche, A., & Uckun, S. (2010). Probabilistic model-based

diagnosis: An electrical power system case study. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *40*(5), 874–885.

Miao, H., Sforna, M., & Liu, C.-C. (1996, Aug). A new logic-based alarm analyzer for on-line operational environment. *Power Systems, IEEE Transactions on*, *11*(3), 1600-1606. doi: 10.1109/59.535703

Misra, A. (1994). *Sensor-based diagnosis of dynamical systems*. Unpublished doctoral dissertation, Vanderbilt University.

Misra, A., Sztipanovits, J., & Carnes, J. (1994). Robust diagnostics: Structural redundancy approach. In *Spie's symposium on intelligent systems.*

Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Systems, Man and Cybernetics*, *29*(6), 554-565.

North American Electric Reliability Corporation. (2012). *2012 state of reliability* (Tech. Rep.). Retrieved from `http://www.nerc.com/files/2012_sor.pdf`

Padalkar, S., Sztipanovits, J., Karsai, G., Miyasaka, N., & Okuda, K. C. (1991). Real-time fault diagnostics. *IEEE Expert*, *6*(3), 75–85.

Poll, S., Patterson-Hine, A., Camisa, J., Garcia, D., Hall, D., Lee, C., ... others (2007). Advanced diagnostics and prognostics testbed. In *Proceedings of the 18th international workshop on principles of diagnosis (dx-07)* (pp. 178–185).

Pourbeik, P., Kundur, P., & Taylor, C. (2006). The anatomy of a power grid blackout-root causes and dynamics of recent major blackouts. *Power and Energy Magazine, IEEE*, *4*(5), 22–29.

Ren, H., Mi, Z., Zhao, H., & Yang, Q. (2005). Fault diagnosis for substation automation based on Petri nets and coding theory. In *Power engineering society general meeting, 2004. ieee* (pp. 1038–1042).

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1996, March). Failure diagnosis using discrete-event models. *IEEE Transactions On Control System Technology*, *4*(2), 105–124.

Sekine, Y., Akimoto, Y., Kunugi, M., Fukui, C., & Fukui, S. (2002). Fault diagnosis of power systems. *Proceedings of the IEEE*, *80*(5), 673–683.

Sun, J., Qin, S., & Song, Y. (2004). Fault diagnosis of electric power systems based on fuzzy Petri nets. *Power Systems, IEEE Transactions on*, *19*(4), 2053–2059.

Taha, W., Brauner, P., Zeng, Y., Cartwright, R., Gaspes, V., Ames, A., & Chapoutot, A. (2012, June). A core language for executable models of cyber-physical systems (preliminary report). In *Distributed computing systems workshops (icdcsw), 2012 32nd international conference on* (p. 303-308). doi: 10.1109/ICDCSW.2012.72

Talukdar, S., Cardozo, E., & Perry, T. (2007). The operator's assistant–an intelligent, expandable program for power system trouble analysis. *Power Systems, IEEE Transactions on*, *1*(3), 182–187.

Tholomier, D., Richards, S., & Apostolov, A. (2007, Aug). Advanced distance protection applications fot dynamic loading and out-of step condition. In *Bulk power system dynamics and control - vii. revitalizing operational reliability, 2007 irep symposium* (p. 1-8). doi: 10.1109/IREP.2007.4410560

Yang, C., Okamoto, H., Yokoyama, A., & Sekine, Y. (1992). Expert system for fault section estimation of power systems using time-sequence information. *International Journal of Electrical Power & Energy Systems*, *14*(2-3), 225–232.

Yongli, Z., Limin, H., & Jinling, L. (2006). Bayesian networks-based approach for power systems fault diagnosis. *Power Delivery, IEEE Transactions on*, *21*(2), 634–639.

Zhang, Y., Ilic, M., & Tonguz, O. (2011, January). Mitigating blackouts via smart relays: A machine learning approach. *Proceedings of the IEEE*, *99*(1), 94 -118. doi: 10.1109/JPROC.2010.2072970

Zhou, G. (1993). A neural network approach to fault diagnosis for power systems. In *Tencon'93. proceedings. computer, communication, control and power engineering. 1993 ieee region 10 conference on* (pp. 885–888).

BIOGRAPHIES

Nagabhushan Mahadevan is a Senior Research Engineer at the Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. His work involves research in model-based diagnostics, verification and validation of health management systems, resilience in cyber-physical systems. He received his M.S. degree in Computer Engineering and Chemical Engineering from the University of South Carolina, Columbia, and B.E.(Hons.) degree in Chemical Engineering from Birla Institute of Technology and Science, Pilani, India.

Dr. Abhishek Dubey is a Research Scientist at the Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. His research interests are related to resilient cyber-physical systems and fault diagnosis in distributed software systems. He received his PhD. in Electrical Engineering from Vanderbilt University in 2009 and B.Tech. in Electrical Engineering from Indian Institute of Technology, BHU, Varanasi, India in 2001.

Dr. Gabor Karsai is a Professor of Electrical Engineering and Computer Science at Vanderbilt University, and Senior Research Scientist at the Institute for Software-Integrated Systems. He conducts research in the design and implementation of cyber-physical systems, in programming tools for model-driven development environments, in the theory and practice of model-integrated computing, and in real-time fault diagnostics. He received his B.Sc., M.Sc., and Dr. Techn degrees from the Technical University of Budapest, Hungary, in 1982, 1984 and 1988, respectively, and his PhD from Vanderbilt University in 1988. Dr. Karsai has worked several large DARPA projects in the recent past: advanced scheduling and resource management algorithms, fault-adaptive control technology that has been transitioned into aerospace programs, and model-based integration of embedded systems whose resulting tools are being used in embedded software development tool chains.

Dr. Anurag K. Srivastava is an assistant professor of electric power engineering at Washington State University and the director of the Smart Grid Demonstration and Research Investigation Lab (SGDRIL). He received his Ph.D. degree in electrical engineering from the Illinois Institute of Technology in 2005. His research interests include power system operation and control using smart grid data. Dr. Srivastava is a senior member of the IEEE, an associate editor of the IEEE Transactions on Smart Grid, and an IEEE distinguished lecturer. He is author of more than 130 technical publications including a book on power system security.

Dr. Chen-Ching Liu is Boeing Distinguished Professor at Washington State University (WSU), Pullman, USA. At WSU, Professor Liu serves as Director of the Energy Systems Innovation (ESI) Center. During 1983-2005, he was a Professor of EE at University of Washington, Seattle. Dr. Liu was Palmer Chair Professor at Iowa State University from 2006 to 2008. From 2008-2011, he served as Deputy/Acting Principal of the College of Engineering, Mathematical and Physical Sciences at University College Dublin, Ireland. Professor Liu received an IEEE Third Millennium Medal in 2000 and the Power and Energy Society Outstanding Power Engineering Educator Award in 2004. In 2013, Dr. Liu was awarded a Doctor Honoris Causa by Polytechnic University of Bucharest, Romania. Professor Liu is a Fellow of the IEEE.

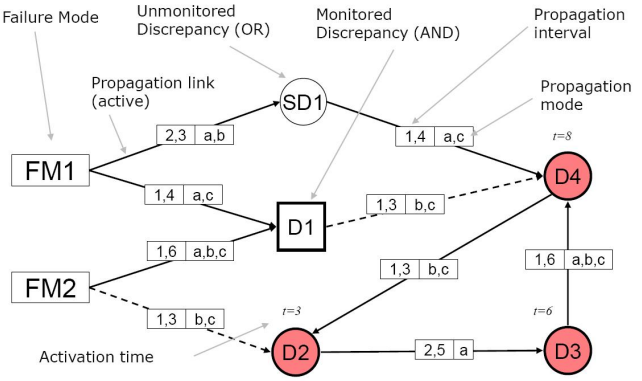


Figure 5. TFPG model ($t = 10$, Mode=A $\forall t \in [0, 10]$.

APPENDIX

A. TIMED FAILURE PROPAGATION GRAPH (TFPG)

A TFPG (Abdelwahed et al., 2004, 2005) is a labeled directed graph. The root nodes are failure modes (fault causes). The other nodes are discrepancies (off-nominal conditions that are the effects of failure modes). Edges between nodes in the graph capture the causality of failure propagation. The edge labels capture the time-interval and operating modes when the failure propagation edge is active. Formally, a TFPG is represented as a tuple $(F, D, E, M, \mathsf{ET}, \mathsf{EM}, \mathsf{DC})$, where:

- F is a nonempty set of failure nodes.
- D is a nonempty set of discrepancy nodes.
- $E \subseteq V \times V$ is a set of edges connecting the set of all nodes $V = F \cup D$.
- M is a nonempty set of system modes. At each time instance t the system can be in only one mode.
- $\mathsf{ET} : E \rightarrow I$ is a map that associates with every edge in E a time interval $[t_{min}, t_{max}] \in I$ that represents the minimum (t_{min}) and maximum (t_{max}) time for failure propagation over the edge.
- $\mathsf{EM} : E \rightarrow \mathcal{P}(M)$ is a map that associates with every edge in E a set of modes in M when the edge is active. For any edge $e \in E$ that is not mode-dependent (i.e. active in all modes), $\mathsf{EM}(e) = \varnothing$.
- $\mathsf{DC} : D \rightarrow \{\texttt{AND}, \texttt{OR}\}$ is a map defining the class of each discrepancy as either `AND` or an `OR` node. An `OR` (`AND`) type discrepancy node will be activated when the failure propagates to the node from any (all) of its parents.
- $\mathsf{DS} : D \rightarrow \{\texttt{A}, \texttt{I}\}$ is a map defining the monitoring status of the discrepancy as either `A` for the case when the discrepancy is active (monitored by an online alarm) or `I` for the case when the discrepancy is inactive (not monitored).

Figure 5 shows a graphical depiction of a failure propagation graph model. Rectangles in the graph model represent

the failure modes while circles and squares represent OR and AND type discrepancies, respectively. The edges between the nodes represent failure propagation. Propagation edges are parameterized with the corresponding interval, $[e.\mathrm{tmin}, e.\mathrm{tmax}]$, and the set of modes at which the edge is active. Figure 5 also shows a sequence of active discrepancies (alarm signals) identified by shaded discrepancies. The time at which the alarm is observed is shown above the corresponding discrepancy. Dashed lines are used to distinguish inactive propagation links.

The TFPG reasoning algorithm attempts to explain the current observations (states of monitored discrepancy nodes) by hypothesizing the faults that could have occured in the system. Each hypothesis assigns a hypothetical state to each node in the graph. In case of failure modes, an ON state indicates that the failure is present, otherwise the state is OFF. The state of a discrepancy node could be set to ON or OFF depending on whether the failure-effect has reached the node or not. Alternately, an UNKNOWN state indicates that there is not enough information to figure out if the failure-effect has definitely reached the node.

The TFPG failure propagation semantics is used to identify and update the hypothetical states of the TFPG nodes. For an OR discrepancy v' and an edge $e = (v, v') \in E$, once a failure effect reaches v at time t it will reach v' at a time t' where $e.\mathrm{tmin} \leq t' - t \leq e.\mathrm{tmax}$. On the other hand, the activation period of an AND discrepancy v' is the composition of the activation periods for each link $(v, v') \in E$. For a failure to propagate through an edge $e = (v, v')$, the edge should be active throughout the propagation, that is, from the time the failure reaches v to the time it reaches v'. An edge e is active if and only if the current operation mode of the system, m_c is in the set of activation modes of the edge, that is, $m_c \in \mathsf{EM}(e)$. When a failure propagates to a monitored discrepancy node (or alarm) v' ($\mathsf{DS}(v') = \mathtt{A}$) its physical state is considered to be ON, otherwise it is considered to be OFF. If the link is deactivated any time during the propagation (because of mode switching), the propagation stops. Links are assumed to be memory less with respect to failure propagation so that current failure propagation is independent of any (incomplete) previous propagation. Also, once a failure effect reaches a node, its state will change permanently and will not be affected by any future failure propagation.

While a detailed description of the TFPG diagnosis algorithm may be found in (Abdelwahed et al., 2004, 2005), in the interest of self-containment a brief description of the procedures referenced in this paper is provided below.

- $Consis(H, O_t)$: This procedure checks if the hypothetical states of nodes as captured in the hypothesis H are consistent with the observations O at time t.
- $UpdateHypo(t, HS_{t-1})$: This procedure takes in as input the current time, t, and the set of hypotheses at the previous time-stamp, HS_{t-1} and outputs an updated set of hypotheses, HS_t which include any updates to the state of the nodes based on the time elapsed.
- $ExplainHypo(H, O_t)$: This procedure generates new hypotheses to explain the current observations (O_t) relative to an existing hypothesis H that explains the past observations.

Case Study: Models for Detecting Low Oil Pressure Anomalies on Commercial Vehicles

Howard E. Bussey[1], Nenad G. Nenadic[2], Paul A. Ardis[3], Michael G. Thurston[2]

[1] *Tixlers Letters, PO BOX 831, Pittsford NY 14534*
howard@tixlers.com
[2] *Golisano Institute of Sustainability, Rochester Institute of Technology, Rochester, NY 14623 USA*
nxnasp@rit.edu mgtasp@rit.edu
[3] *GE Global Research,1 Research Circle Bldg. K1-4A6, Niskayuna, NY 12309*
ardis.p@ge.com

ABSTRACT

We present a case study of anomaly detection using commercial vehicle data (from a single vehicle collected over a six-month interval) and propose a failure-event analysis. Our analysis allows performance comparison of anomaly detection models in the absence of sufficient anomalies to compute the Receiver Operating Characteristic curve.

Several heuristically-guided data-driven models were considered to capture the relationship among three main engine signals (oil pressure, temperature, and speed). These models include regression-based approaches and distance-based approaches; the former use the residual's z-score as the detection metric, while the latter use a Mahalanobis distance or similar measure as the metric. The selected regression-based models (Boosted Regression Trees, Feed-Forward Neural Networks, and Gridded Regression tables) outperformed the selected distance-based approaches (Gaussian Mixtures and Replicator Neural Networks). Both groups of models were superior to existing Diagnostic Trouble Codes. The Gridded Regression tables and Boosted Regression Trees exhibited the best overall metric performance.

We report a surprising behavior of one of the models: locally-optimal Gaussian Mixture Models often had zero detection performance, with such models occurring in at least 25% of the iterations with seven or more Gaussians in the mixture. To overcome the problem, we propose a regularization method that employs a heuristic filter for rejecting Gaussian Mixtures with non-discriminative components.

Howard Bussey et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION AND BACKGROUND

Equipment health and condition monitoring enables maintenance to minimize the effects of equipment degradation or failure. Building on existing concepts for predictive maintenance, Reliability Centered Maintenance (RCM) (Nowlan & Heap, 1978) provided a formalism for Condition-Based Maintenance (CBM). Being based upon objective evidence of equipment degradation or impending failure, CBM has significant economic and safety benefits: it reduces incidence of unscheduled failures and downtime and reduces occurrence of unnecessary or early scheduled maintenance.

Health or condition monitoring is the process of collecting asset data, extracts the information and provides it to CBM. Affordable sensors, data storage, and networking enable comprehensive monitoring of all types of assets. In order to make this data actionable for CBM, models are needed to identify and characterize anomalies, and then to relate the anomalous patterns to forward looking failure risk for decision making purposes (prognostics). The models are typically classified as expert-system, physics-based, data-driven, and hybrid. This paper takes the data-driven modeling approach.

Health monitoring is generally an incremental (not all-at-once) process, as data is typically not available to develop comprehensive diagnostic and prognostic algorithms from the outset (Sikorska, Hodkiewicz, & Ma, 2011). Most modern vehicles are equipped by the original equipment manufacturer with built-in sensors on a data bus, and diagnostic systems that detect major drive train failures. The diagnostic coverage on these systems can be limited, and they typically detect problems with limited warning horizon before maintenance action is required. Telematics systems, such as General Motors OnStar™are increasingly being used to monitor private, commercial, and military vehicles. Data provided by these systems, over a large fleet of vehicles, can be used to develop new anomaly detection and failure prediction al-

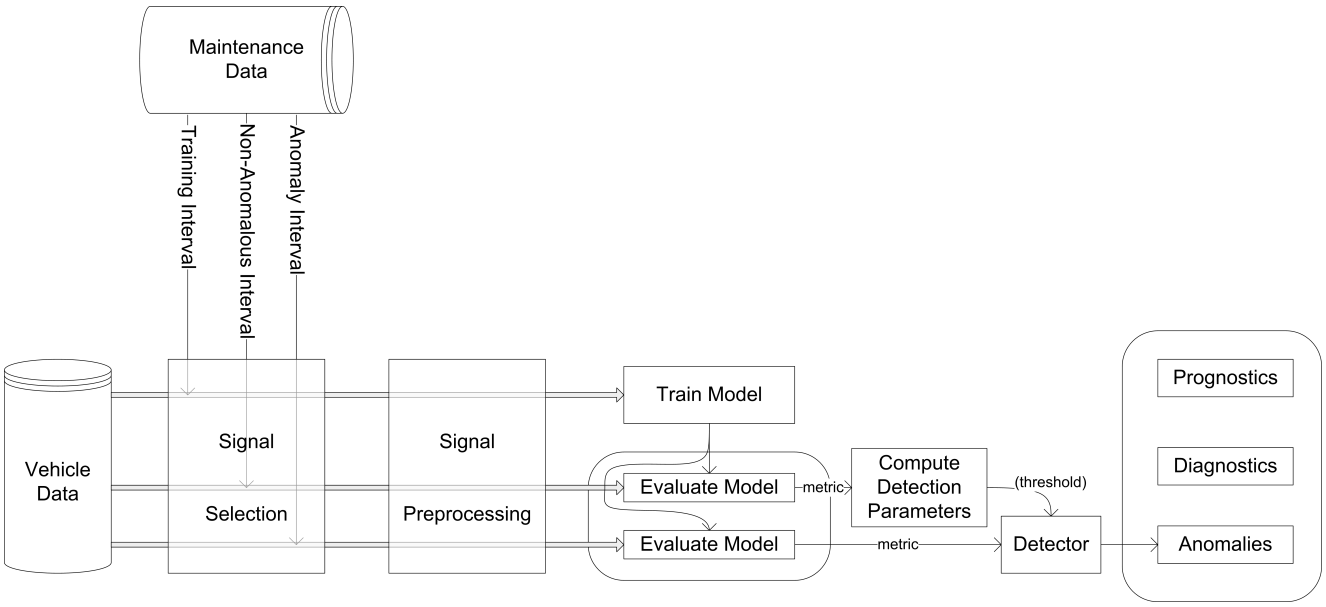


Figure 1. Analysis process, showing steps of building the model, detection anomalies, diagnosing faults, and predicting future failures (prognostics).

gorithms more cost effectively than through traditional engineering testing. On-board computers, coupled to the vehicle data bus, can filter vehicle data and run algorithms locally, or they can relay data to a back-end system for processing. These systems can also support cost effective addition of vehicle sensors to augment existing capabilities. In addition to driver services and logistics support, these systems are used to collect information to support product improvement, and have growing levels of Prognostic Health Management (PHM) capability.

Consolidation of vehicle fleet data in a data warehouse provides an opportunity to develop CBM knowledge and algorithms incrementally. As failures occur within the fleet, the vehicle and maintenance data can be correlated, analyzed, and used to create autonomous health monitoring agents with embedded anomaly detection, diagnostics, and prognostics. With larger fleets, more accurate and extensive algorithm sets can be developed. Our approach is opportunistic, based upon the failures, and data-driven, exchanging data mining and statistical machine learning in place of in-depth expert knowledge.

As shown in Figure 1, anomaly detection is the first layer of information extraction in condition-based maintenance. The ability to reliably detect system performance changes, in the context of different operating and environmental conditions, is the first step towards condition monitoring. The value of anomaly detection is the ability to trigger useful alerts and to pave way to more sophisticated PHM. In the context of truck fleet operations, an anomaly warning can be provided to maintenance or operational supervisors to prompt them to review the condition of the truck or the behavior of the driver.

Observed anomalies and their links to the associated failure modes (established by maintainers) form a labeled data set suitable for supervised machine learning. Automated classification of observed anomalies enables the second level of PHM – diagnostics. Using observations of operational failures for classification training is well suited for environments where failures can be, or have historically been, tolerated; this approach is cost effective and requires no additional risk. In particular, the present case study is concerned with health monitoring of commercial truck fleets, where failures can be very costly, but are tolerated as a part of doing business. The variant of this approach, in which unsupervised anomaly detection identifies candidate events for human expert analysis, may be suitable for systems such as nuclear reactors where system failures are unacceptable. In this case, the data-driven approach would augment the physics- or expert-knowledge-based systems presently in use. This paper focuses on the development of a methodology for anomaly detection in truck engine behavior using data captured from a commercial fleet telematics system. To achieve this capability, we use data-driven models, each with an intrinsic metric. We will describe five such models, motivate their choices, and compare their performance in following sections.

Once anomaly detection is in place, additional observed failures can be used to improve anomaly detection algorithms and parameters, as well as to develop diagnostics and prognostics. The development of data-driven prognostics is enabled (and improved) by more examples of the same failure mode, which allow for the development of models of the pro-

gression of failures subject to operational and environmental context (regression, tracking). Alternatively, correctly classified anomalies with accurate physics-based (or other expert knowledge-based) models can be considered without requiring a large number of examples. Data-driven diagnostic development is enabled by examples of a variety of distinct failure modes; from a machine learning viewpoint, diagnostics can be perceived as a discrete classification problem. Since the available data have only one failure, we were unable to address the diagnostic and prognostic areas.

Building a system for anomaly detection includes the following three steps: 1) selecting and pre-processing the relevant signals; 2) selecting, building, and tuning a model equipped with a metric; and 3) selecting and tuning an inference engine that indicates anomalies, based upon the model metric. While design, parameterization and parameter tuning of all three blocks impact the performance of the system, this report focuses on model selection and tuning. In all cases the models operate on the same three signals: engine oil temperature, engine oil pressure, and engine speed. Moreover, all systems discussed in this paper employ a simple inference engine a low-pass filter followed by a comparator. When the filtered metric exceeds the threshold, the signals are considered anomalous.

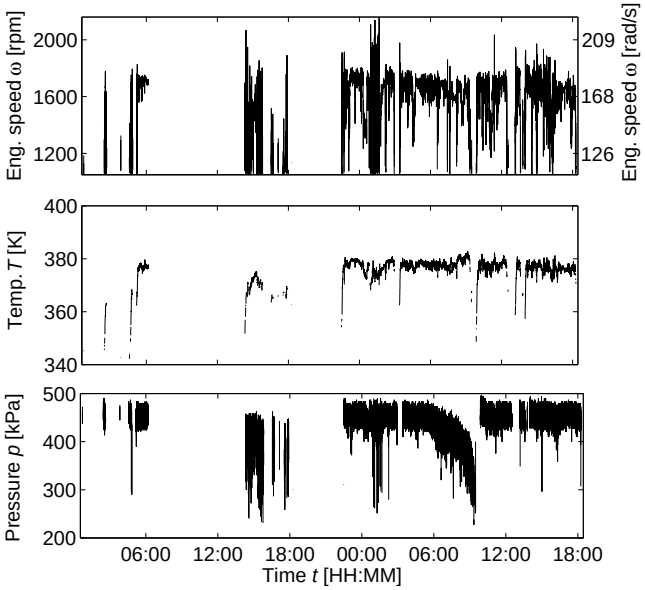


Figure 2. Example engine speed, oil temperature, and pressure

Chandola, Banerjee, and Kumar (2009) survey anomaly detection techniques, touching on methods used here. Our work falls under their industrial damage classification, for which they report on work using parametric and non-parametric statistical modeling, Neural Networks, spectral, and rule-based systems. Bishop (2006) describes these machine learning techniques in further detail, including specifics of training and testing that are used in our work. Vachtsevanos, Lewis, Roemer, Hess, and Wu (2006) present a somewhat different model for data-driven anomaly detection (fault or failure detection in their terminology - see their section 5.2.3). The literature reporting anomaly detection results using standard vehicle data over long periods is sparse. Golosinski, Hu, and Elias (2001) report on 1.2 hours of data from a single vehicle. Kargupta et al. (2004) report on analysis based upon a vehicle simulator. McArthur, Booth, McDonald, and McFadyen (2005) report on a processing system using data from a single engine. Cheifetz, Same, Aknin, and de Verdalle (2011) report on data from 22 consecutive operating cycles of a commercial bus. Our experiments are intended to provide further empirical insight, especially with regard to longer performance periods and the specifics of model construction.

The study data include a period during which the vehicle was driven with an active oil leak. We employed an opportunistic data-driven methodology in our analysis. Because we have only one labeled failure event in the data, we: (a) create several models from the training data; (b) for each model, find the minimum threshold that results in a zero false-alarm rate during the normal period; (c) measure detection performance during the low-oil period using the models and their respective detection threshold values. For this failure, we have approximately 144 hours of training data from a two-week interval, failure data representing about 15 hours of operation during approximately 19 clock hours, and the normal period of five months (1500 hours) following repair.

2. PROBLEM AND PROCESS

Figure 2 shows a segment of the vehicle data: engine speed and oil temperature and pressure, recorded over a two-day period during which the vehicle was operated with an oil leak. The data show the vehicle operating with steadily declining oil pressure starting between 5:30 and 6:00 AM. With this rich contextual information, one can conclude that the pressure is legitimately anomalous. However, if only the pressure information is available, the most one can say is that the pressure exhibits a downward trend. For this fault, anomaly detection based upon only the oil-pressure is insufficient. The manufacture recommends pressures of at least 150 kPa when the engine is idling, and at least 300 kPa when the engine speed is greater than 1100 RPM. If anomaly detection used only the idle condition minimum pressure, the anomaly would be missed in its entirety. Using the higher limit, the anomaly is detected only in the last few minutes, and might cause false alarms if applied when the engine is idling. Some anomaly detection algorithms use a mode-based approach, where the operating modes and associated signal limits are defined a priori and used to identify anomalous operations. Based on the rules presented above, a mode-based oil pressure anomaly detector would identify anomalies sometime after 9 AM on the

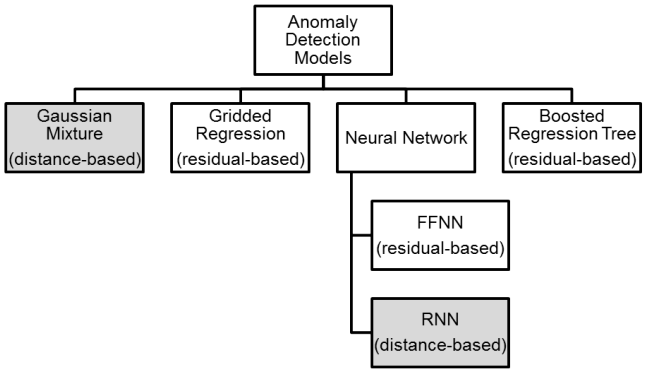


Figure 3. Anomaly detection approaches in this investigation

second day of second day of operation.

Figure 3 maps the five models this study explored: three residual-based models - Gridded Regression table (GR), Boosted Regression Tree (BRT), and Feed-Forward Neural Network (FFNN) - and two distance-based models - Gaussian Mixture (GMM) and Replicator Neural Network (RNN). In the residual-based systems, the models predict the pressure, based upon temperature and engine speed. The metric is the absolute value of the z-score (*the standard score*) of the residual, where the residual mean and variance are determined from the model and training data. In the distance-based systems, the metric reflects how different all three signals are from the model.

2.1. Data Source and Preparation

As indicated in Figure 1, signal preprocessing is often necessary before the data is used for building models. The preprocessing here includes filtering out irrelevant data (e.g. during idling), removal of short-duration transient data, eliminating non-informative data (e.g., if some data is missing), and excluding data segments so short they cannot be handled in subsequent processing (e.g., a 20 s drive between two 5 minute idle periods).

We use data from a commercial truck (including both maintenance data and operational data from the vehicles' data buses) as provided to RIT by Vnomics Corp. Examination of the maintenance data showed that there was one oil leak event; that single event is used as the fault event for this study. The vehicle data were obtained from J1587 and J1939 packets available on the J1708 and CAN buses on heavy-duty trucks. This data did not include oil level information, even though that signal is defined in the J1939-71 and J1587 specifications. The Vnomics' Vehicle Health Management Software (Vnomics, 2012) collected the asynchronous on-board signals and used lossy data compression to save space in the database. The compression algorithm compares the current signal value to the last stored data value and stores the current signal value if the difference exceeds a fixed threshold. The thresholds are provided in Table 1.

Table 1. Thresholds used in data compression algorithm.

Signal	Threshold
Oil Temperature	0.2 C/K
Engine Speed	10 RPM
Oil Pressure	6.89 kPa

For this investigation, the asynchronous signal values are read from database and time-synchronized to a 1 s periodic stream using sample and hold interpolation. In addition to synchronization, some data are removed. For instance, we remove low-RPM (idle) data so that it isn't over-emphasized during training. There are two irrelevant data removal schema, as show in Table 2. In schema 1, a wide range of physically-feasible engine oil temperatures are accepted. In schema 2, the temperature range is narrower to exclude data collected while the engine is warming up.

Table 2. Data Removal Schema.

Schema	Signal	Minimum (inclusive)	Maximum (exclusive)
1	Temperature	-20	120
	RPM	1050	2500
	Pressure	50	550
2	Temperature	90	120
	RPM	1050	2500
	Pressure	50	550

The training interval was selected after inspection of the operational and maintenance to find the first period with no maintenance events and no obvious data anomalies. For this vehicle, that was immediately following a stuck at high oil pressure sensor fault. The selected training period, with approximately 142 hours of operational data, is the two weeks following replacement of the sensor. After removing irrelevant data, there remain 75 hours of training data. The non-anomalous period follows the repair of the oil leak. The anomalous period is a two-day period starting 2/2/2010.[1]

2.2. Metric Filtering

For all of these models, the metric is filtered with an infinite impulse response low-pass filter with low passband frequency of 0.0017 Hz (1/600 Hz) and reject-band frequency of 0.05 Hz. These values were chosen to provide a filter time-constant of 5 minutes. This filter is appropriate for detecting anomalies related to a slow oil leak.

In addition to the low-pass filtering, the metric filtering must deal with data gaps introduced by the irrelevant data removal step described in 2.1. In addition, short segments (e.g. 60 s) are statistically insignificant when a fault event evolves over a period of an hour or longer; because they cause numerical instability, we removed them. Finally, the filter used above is applied on the remaining segments on a segment-by-segment

[1] To encourage further research in this area, we have made the data available: http://www.rit.edu/gis/research-centers/csm/EOP_Case_Study.php. This has irrelevant data discarded according to schema 1.

basis. This filter exhibits some ringing, so to prevent high amplitude ringing, the filter is initialized with 10,000 s of input points equal to the median of the first 50 samples in the segment.

Because our goal is to study performance of several system models, the same data preparation and detection processing steps are used for all of the models.

2.3. General Modeling Process

For a problem of this type, the inputs consist of n observed signals $S_1, S_2, \ldots, S_n$. Data is divided into training $D_{training}$, event D_{event}, and normal D_{normal} sets, such that the sets are subsets of $\mathbb{R}^n$:

$$D_{training}, D_{event}, D_{normal} \in \mathbb{R}^n \tag{1}$$

and the sets are disjunct

$$\begin{aligned} D_{training} \cap D_{event} &= \varnothing \\ D_{training} \cap D_{normal} &= \varnothing \\ D_{normal} \cap D_{event} &= \varnothing \end{aligned} \tag{2}$$

The modeling is the process of identifying parameters of a model $\mathcal{M}$ and detection threshold Θ, given metric m, that maximizes discriminability between the training and event data:

$$\max |m(\mathcal{M}(D_{training}), \mathcal{M}(D_{event})) > \Theta| \tag{3}$$

subject to zero false alarms

$$|m(\mathcal{M}(D_{training}), \mathcal{M}(D_{normal})) > \Theta| = 0 \tag{4}$$

Overall, our goal is to provide a long and stable detection horizon for known faults, subject to the requirement that there are no anomalies detected during the normal interval (false alarms). As a final note, we prefer low-complexity models that use zero expert system knowledge and have short training times.

All five models, described in the next section, were able to detect anomalies on the first day of the low-oil event, which took place approximately 19 hours before the last mission during the low oil period. Analyzing these anomalies showed transient pressure drops when the engine speed briefly increased to a range between1500 and 2000 RPM. Figure 4a, from the training interval, shows a small pressure variation, approximately 50 kPa, with no clear pattern of increasing or decreasing. Figure 4b shows the data from one of the anomalous intervals. Here the pressure drops approximately 100 kPa as the engine speed increases from 1500 to 2000 RPM. In both cases, the pressure is above 400 kPa when the engine speed is steady around 1500 RPM.

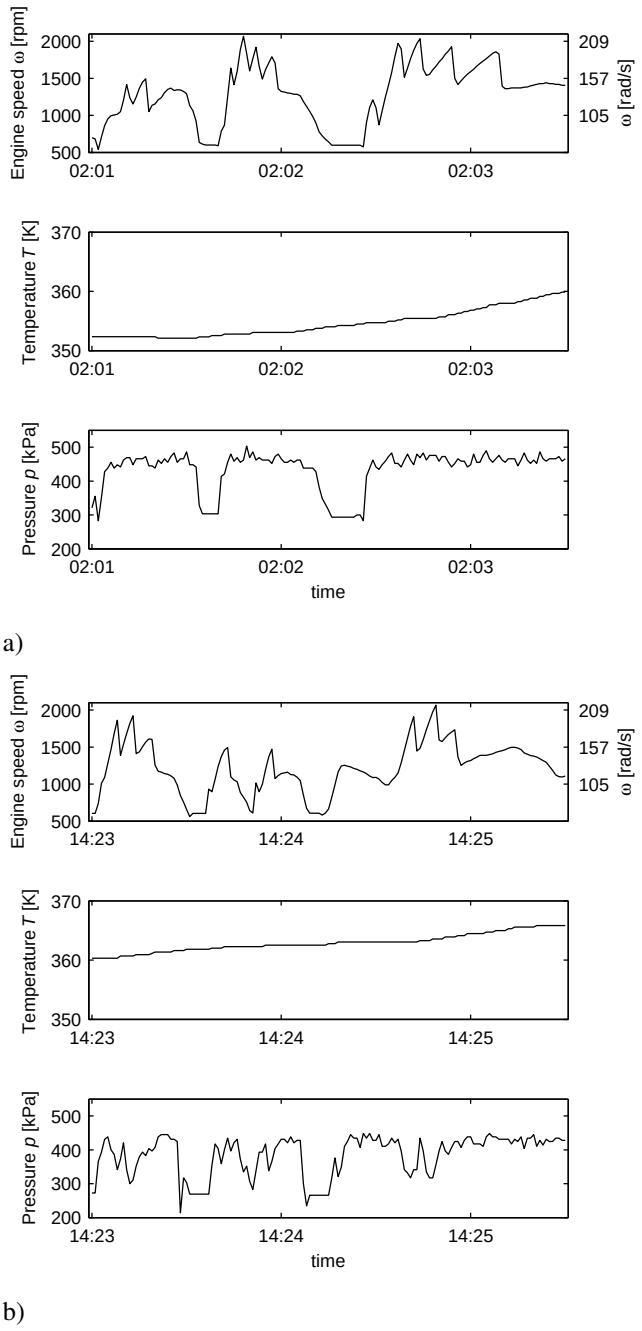


Figure 4. (a) Signals on the first training day (1/14/2010) showing the normal behavior where engine speed spikes make little change in the oil pressure. (b) Anomalous signals at 14:23 on first day of low oil event (2/2/2010), where the pressure drops to approximately 325 kPa when the engine speed increases sharply from 1500 RPM to 2000 RPM - once just after 14:23, and again just before 14:25.

3. MODELS' DESCRIPTIONS AND PERFORMANCES

This section describes the five models in turn, with the application-specific decision processes associated with the models and

their performance.

3.1. Model 1 – Gridded Regression

The Gridded Regression (GR) model has a look-up table used to estimate engine oil pressure p as a function of engine speed ω and engine oil temperature T; and the residual mean and variance, used to calculate the z-score metric. Here, the domain, the temperature-speed (ω-T) plane, is subdivided into rectangular subdomains, or bins, as depicted in Figure 5a. The temperature and speed ranges are determined a priori, based on the expected ranges of the signals; consequently, some of the bins are empty during training. The discrete pressure estimates $\hat{p}$ over the domain are given by

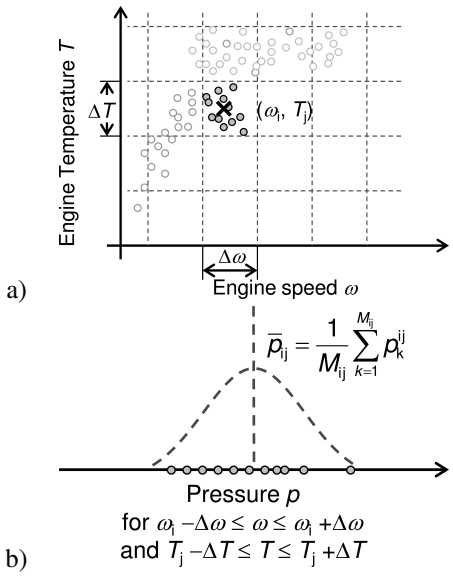


Figure 5. A sketch of GR model. (a) Discretized (ω-T) plane. Data points within (ω_i-Tj) bins are highlighted. (b) Mean pressure of the data.

$$\hat{p} = f(\omega, T) = \overline{p}_{ij} \tag{5}$$

where f is the point sample of a 2-dimensional Gaussian distribution in terms of ω and T. $\overline{p}_{ij}$ is the mean pressure of the training data corresponding to (ω_i-T_j) subdomain bounded by $\omega_i - \Delta\omega/2 \leq \omega < \omega_i + \Delta\omega/2$ and $T_j - \Delta T/2 \leq T < T_j + \Delta T/2$ (see Figure 5b), as in:

$$\overline{p}_{ij} = \frac{1}{M_{ij}} \sum_{k=1}^{M_{ij}} p_k^{ij} \tag{6}$$

Another way to think of this model is a piece-wise constant (in this case two-dimensional) fit function with error bars. In the metric evaluation operations, subtracting estimates from the measurements yields error $\varepsilon_p = p - \hat{p} = p - \overline{p}_{ij}$. The residuals are considered collectively, over all bins. The metric used for detecting anomalies is the absolute value of the z-score of the residuals, computed as:

$$m = |z_p| = \left| \frac{\varepsilon_p - \overline{\varepsilon_p}}{\sigma_p} \right| \tag{7}$$

Figure 6a shows that the Gaussian distribution fits the residual data, $\varepsilon_p \sim N(0, \sigma_p^2)$, reasonably well. Figure 6b quantifies this fit further, showing that 99.8% of the residuals match the expected range from -25 to +25.

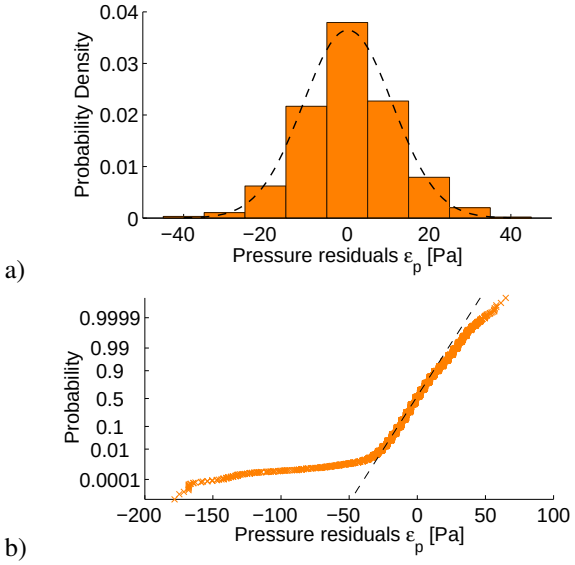


Figure 6. Distribution of the 75 hours of training data. (a) Histogram with a fit. (b) Test of normal data.

3.1.1. GR: Parameters and Performance

The oil temperature and RPM ranges were divided into 10 equal intervals, resulting in a 10x10 grid. The model estimate for each bin in the grid is the mean oil pressure for the data samples in that bin. If the count of data in the bin was too low, the model estimate for that bin was NaN (not-a-number) a flag value causing that bin to be effectively ignored in the rest of the experiment. The residuals were computed over all of the training data, and the histogram of the residuals in Figure 6 shows the distribution is well-modeled by a Gaussian distribution. The variance of the residuals is computed and stored with the model, to be used in subsequent z-score calculations. For each data point in the test and non-anomalous intervals, the GR model is used to predict the oil pressure, based upon the RPM and oil temperature. The metric is the absolute value of the z-score of the residual. The metric is smoothed by the low-pass filter described in Section 2.2. For the non-anomalous interval, the smoothed metric value is used to determine the detection threshold, guaranteeing the no false alarm criterion. That threshold is compared with the smoothed metric for the test interval, and the results are shown in Figure 7. The anomalies between 15:00 and

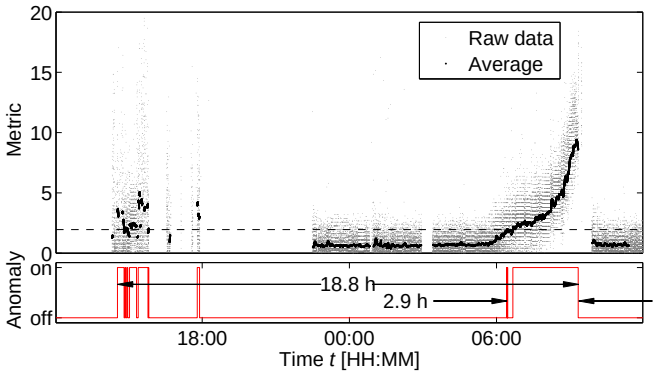


Figure 7. Performance of GR Model. Detection horizon is about 2.9 hours.

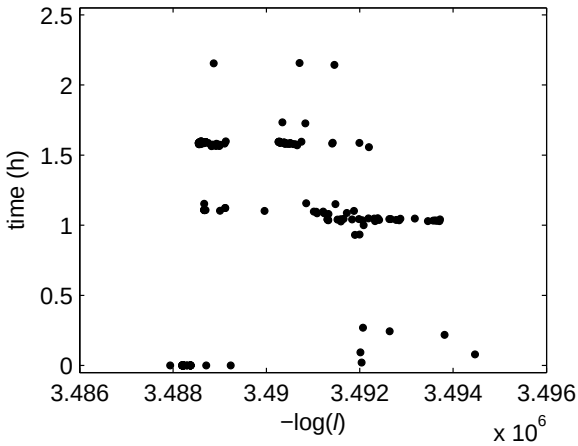


Figure 8. Performance of GMM(7)s. Each dot on the figure represents one trained GMM(7). The models with the better likelihood generally have better detection performance, although the models with the best likelihood have zero detection performance.

18:00 are correlated with vehicle oil level and pressure vehicle Diagnostic Trouble Codes (DTCs) recorded at 14:15 and 15:38; however, they are not included in the detection horizon calculation, which is based upon the period between 22:35 on day 1 and 09:21 on day 2 of these data. This narrower time range is used because the vehicle operators, aware of the oil leak, added oil from time to time in this period. However, the period from 22:35 until 09:21 the next morning, as Figure 2 shows, represents a single event when the oil pressure dropped from normal to abnormally low.

Tuning this model requires selection of the number of bins for temperature and engine speed. The number we used represents a compromise between too few bins, which would increase the prediction error, and too many bins, which would result in too few training points per bin. Given the bin count selection, training is deterministic for a given training data set.

The selection of 10 bins was based on trial and error in this study. Optimal or near-optimal bin counts could be selected through either exhaustive or random exploration of the bin count space for each independent variable.

3.2. Model 2 – Gaussian Mixtures

Model 2 is an automatically trained GMM comprising a set of multivariate normal distributions, $\boldsymbol{N}_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, and their weights π_k where $\sum \pi_k = 1$. The distributions, $\boldsymbol{N}_k$, are trained to maximize the generative likelihood of all points (T_t, ω_t, p_t) in the training data. The metric used in this model is the likeliest Mahalanobis distance (Duda, Hart, & Stork, 2000), which is the Mahalanobis distance to the mean of the Gaussian G_k that maximizes $\mathcal{P}_t = pr_k(T_t, \omega_t, p_t) \cdot \pi_k$.

Two variants of the model were considered: one (schema 1) explored wide temperature range and the other (schema 2) was restricted in a narrower temperature range.

3.2.1. GMM: Parameters and Performance

The modeled employed seven fitted Gaussian distribution mixture components. The number of components was determined heuristically by searching the parameter space between one and 15 Gaussian components in the GMM: mixtures with less than seven components exhibited shorter detection horizon, while mixtures with more components showed no consistent advantage in detection horizon, and sometimes resulted in a large proportion of models with zero detection performance. Candidate GMMs were trained with Matlab® using the `gmdistribution.fit()` method. This uses an expectation maximization algorithm to find locally optimal models meeting hard-coded convergence criteria.

Initial experiments showed inconsistent performance with detection horizons ranging from 0 to 2.2 hours (see Figure 8). The cause for this is explained in section 3.2.2. The results shown in this section use models trained with the combined expectation maximization and rejection criterion filter. The metric performances similar to the one in Figure 7, and are not repeated for each of the models for brevity.

Changing the irrelevant data removal to schema 2 and rerunning the same experiment resulted in no performance improvement, showing that the GMM training and rejection filtering process is robust in that the detection horizon is the same for two different temperature ranges. While the horizons are the same (see the GMM(7) schema 1 and schema 2 results in the figure), the schema 2 results, based upon data in a narrower temperature range, show less variation at the onset of detection (07:40) on the second day.

Training required repeated creation of GMMs from different random subsets of the training data, with selection of

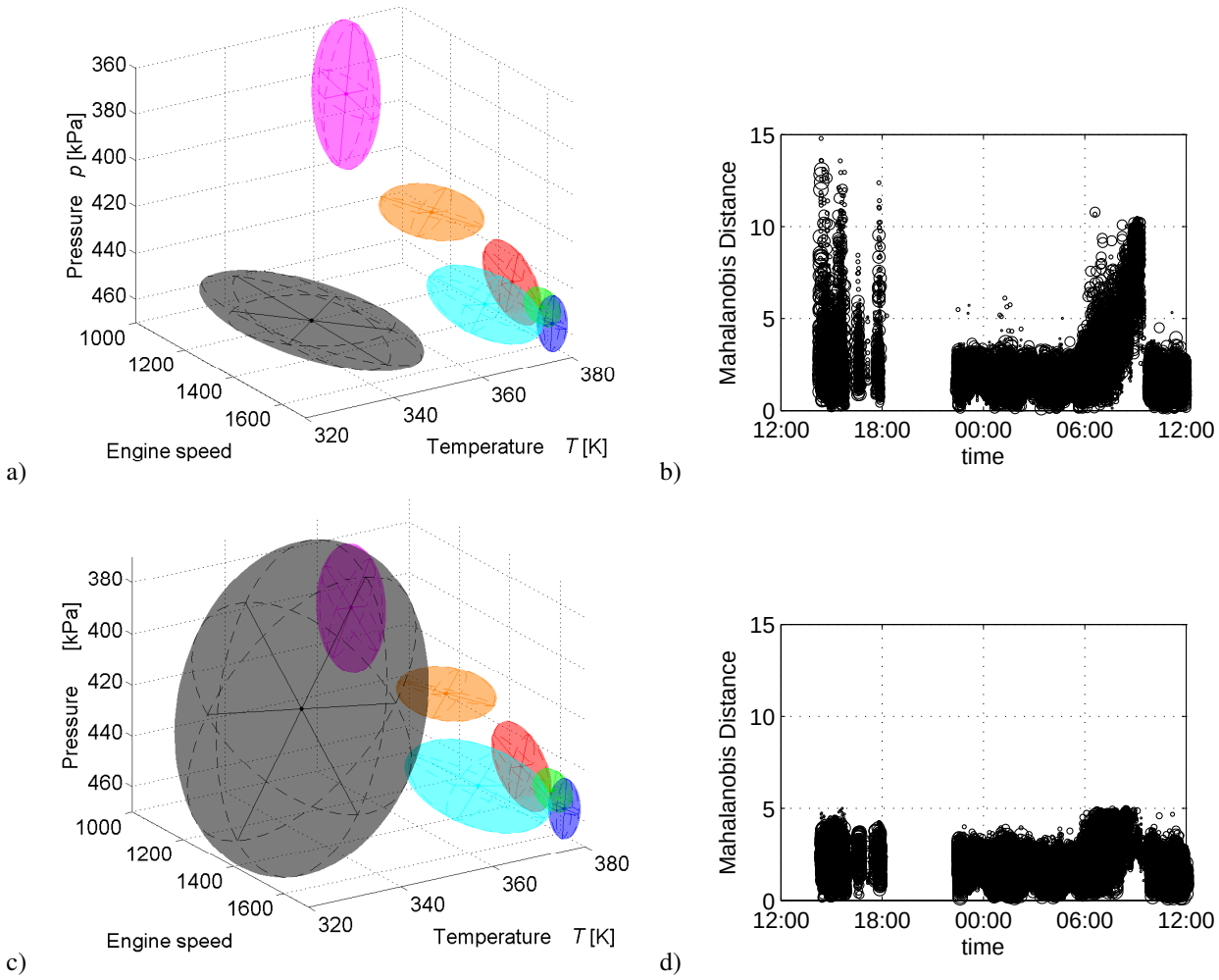


Figure 9. Visualizations of a GMM. (a) with good (1.6 h) prediction horizon; (b) GMM with zero prediction. Most of the Gaussians (except the grey one) have similar positions and sizes as the ones in (a).

the GMM with the smallest average Mahalanobis distance to the most likely Gaussian for all the training data. The number of Gaussians in the GMM was selected by searching for the smallest number of components where the improvement of the average Mahalanobis distance stopped to avoid overfitting.

3.2.2. Gaussian Mixture Rejection Filtering

We investigated observed inconsistency in performance of randomly-initialized GMMs in order to understand why some resulted in zero detection performance. Figure 8 shows the relationship between the model performance and the likelihood, l, of the training data given the trained model for GMMs with seven components each. The figure shows that several of the learned models those with the best training performance have zero detection. The results for the other GMMs show a general correlation between training performance (larger model posterior likelihood, l, or smaller $-\log(l)$) and detection performance. The GMM visualizations in Figure 9 – one with 1.6 hour detection horizon and one with zero performance – show the likely cause of this. (The ellipsoids represent the envelope enclosing the points within the one standard deviation probability, that is where $|z| \leq 1$.) In the GMM with good performance, Figure 9a, the component Gaussians are all fairly compact. The other, Figure 9c, shows that one of the Gaussians encloses a large volume of the $[T_t, \omega_t, p_t]$ space. With this model, the metric values are all less than 5.

The GMMs like the one shown in Figure 9c are non-discriminative. The most likely Mahalanobis distances of any point in the training, anomaly, or post repair data set, is small enough that no anomalies are detected according to the problem statement in section 2.3. Figures 9b and 9d show the likeliest Mahalanobis distance of the low-oil interval data,with respect to the clusters of the two models shown in Figure 9a and Figure 9b, respectively. In a more detailed examination of the results, we found that the maximum Mahalanobis distance of any point

in the training data to the large-ellipsoid component of Figure 9 (or any of the ones with zero detection performance) was less than 7. Based on this, the rejection criterion used to reject GMMs with non-discriminative Gaussian components is *for each Gaussian component in the GMM, compute the Mahalanobis distance between that Gaussian and each point in the training data. Reject the GMM if the maximum Mahalanobis distance for any component is than a threshold.* For this study, the rejection threshold value was 10. This value must be selected, based on the performance of the trained GMMs, by comparison of results of several GMMs with reasonable detection horizons with several GMMs with zero or near-zero detection horizons.

We applied this criterion to 20 candidate GMMs; 7 (35%) were rejected. We selected the GMM for modeling from the remaining GMMs by finding the GMM with the highest likelihood of the training data. The GMM with the longest detection horizon (see Figure 8) 2 hours did not have the highest likelihood. That model could not be selected according to the rules presented in the problem statement (section 2.3) because it used data other than the model and the training data.

We tested the need for this rejection filtering by using the algorithm of (Figueiredo & Jain, 2002) for training GMMs. We found a clear threshold for the rejection criterion after training 120 different GMMs. We found that GMMs that were rejected had zero detection performance. Although this alternate means to train GMMs confirmed the need for rejection filtering, and has several advantages over the native Matlab method especially finding the optimal number of components in the GMM we did not use this algorithm for the work reported here because the GMMs trained with this algorithm did not perform as well as the ones trained by Matlab's `gmdistribution.fit()` method.

3.3. Model 3 – Feed-Forward Neural Network

Two Artificial Neural Network (ANN) models were explored. The first one, Feed-Forward Neural Network (FFNN) can be viewed as a neural network analogue of Gridded Regression. An FFNN was trained to estimate the engine oil pressure, given the oil temperature and the engine speed. A new unknown function f_{NN} is trained to express pressure in terms of the other two variables and unknown parameters – weights $\mathbf{w}$

$$\hat{p} = f_{NN}(T, \omega; \mathbf{w}) \tag{8}$$

The metric used was the same as for the GR model: the absolute value of the z-score of the residuals. The hidden neurons employ sigmoid activation functions because linear activation functions reduce the neural network to a simple linear equation

$$\hat{p} = w_0 + w_1 T + w_2 \omega \tag{9}$$

whose performance was considerably worse than that of the GR model.

3.3.1. FFNN: Parameters and Performance

At first, a two-layer neural network was employed for modeling the functions[2], with twenty neurons in the hidden layer, given by

$$\hat{p}(T, \omega; \mathbf{w}) = \sigma\left(\sum_{j=1}^{40} w_{kj}\sigma\left(w_{j1}T + w_{j2}\omega + w_{j0}\right) + w_0\right), \tag{10}$$

where $\sigma()$ is the logistic sigmoid and $\mathbf{w}$ are the weights. This standard neural network topology, known as the universal function approximator, with its expressive power, and its relation to Kolmogorov theorem is discussed in (Duda et al., 2000, Section 6.2.2). However, in our case, significantly better performance was achieved after the two-layer topology 2-20-1, was replaced by a three-layer 2-3-3-1 topology with the same total number of neurons, which is not surprising because deeper network have better expressive power. The final topology was selected comparing various candidate topologies. The number of layers and neuron counts were randomly selected within narrow ranges. A simple program trained FFNNs with the selected topology and evaluated the event horizon. The best model, with the longest event horizon, was used. Figure 10 shows the topology of a six-neuron FFNN. This simple network performed strictly as well, or better than, FFNNs with larger numbers of neurons or additional neuron layers. The selected topology was simplest in terms of neuron counts, and is expected to have better generalization than its more complex counterparts

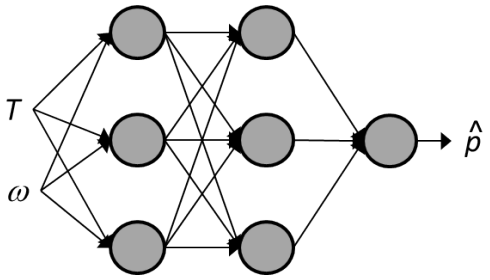


Figure 10. Topology of FFNN – two hidden layers with three neurons each.

After training on good data, the network showed a 3 hour detection horizon with no false alarms.

3.4. Model 4 – Replicator Neural Network

The second ANN model, Replicator Neural Network (RNN), can be considered as neural network analogue of GMM. An RNN (Hawkins, He, Williams, & Baxter, 2002) has 3 hidden layers, with sigmoid activation functions in the first and third

[2]This article employs the notation where the number of layers of a neural network is equal to the number of adaptive weights, as in (Bishop, 2006).

layers. The middle hidden layer has one neuron for each input signal, and the activation function is a differentiable step function that quantizes the input into one of the steps. The output of the network is the vector $[\hat{T}_t, \hat{\omega}_t, \hat{p}_t]$.

We found that the RNN model did not train well with the original input data; the average length of the residual vector was dominated by prediction error of ω. This necessitated scaling the training, anomaly event, and normal data. For the metric, Hawkins et al. (2002) suggests using the outlier factor, which is defined as the mean of the square of the Euclidean-norm of each residual:

$$OF_t = \frac{1}{3}((T_t - \hat{T}_t)^2 + (\omega_t - \hat{\omega}_t)^2 + (p_t - \hat{p}_t)^2) \quad (11)$$

We also investigated an alternative metric: the Mahalanobis distance of the residuals from the mean of a single Gaussian modeling the residuals from the training interval. The outlier factor weights all components of the residual equally, whereas the Mahalanobis distance metric adapts to the statistics of the residual signals.

3.4.1. RNN: Parameters and Performance

An RNN was trained to replicate T, ω, and p. The signal values were pre-scaled into the range [0.1 0.9]. Mahalanobis distance metric resulted in a 1.2 hour detection horizon. Hawkins' (Hawkins et al., 2002) outlier factor metric resulted in zero anomaly detection.

Guided by an automated exploration of the parameter space, we selected a RNN with 10 neurons in the first and last hidden layers, and 3 neurons in the middle hidden layer, corresponding to our three signals in this study. The activation function of the middle hidden layer has 32 steps.

Mahalanobis distance metric resulted in a 1.2 hour detection horizon.

3.5. Model 5 – Boosted Regression Tree

The BRT (Elith, Leathwick, & Hastie, 2008) model estimates $\hat{p}_t$ based upon (ω_t, T_t). From the modeling perspective, it is comparable to the GR model because both use speed and temperature to predict the pressure, then calculate the absolute value of the z-score given by Eq. (7) as the metric.

3.5.1. BRT: Parameter and Performance

A BRT, with 200 sub-trees, was trained on data with range filtering according to schema 1. The detection horizon was 2.9 hours, as shown in Figure 11. We trained models with 10, 20, 50, 100, and 200 sub-trees, and found that the performance for the 10 sub-tree BRT was much lower (1.2 hours), while the BRTs we investigated with 20 – 200 sub-trees all produced detection horizons within 0.1 hours of each other.

In another variation on this experiment, we used data using schema 2 for the range filter (restricted oil temperature) and found that performance improved substantially: for the 20, 50, 100 and 200 sub-tree BRTs, the detection horizon was 3.0 hours, and the detection horizon of the 10 sub-tree BRT was only slightly less - 2.8 hours.

4. RESULTS COMPARISON

Figure 11 and Table 3 summarize the results of this investigation. Figure 11 offers two comparisons based on two detection horizons: one measures the time between the first observed anomaly (the day before the final failure) and the final failure, and the other measures the time between the first detection of anomaly during the final mission and the final failure.

The performances of the detectors according to the first, accross-the-mission comparison are nearly indistinguishable, ranging between 18.8 and 19 hours, which amounts to just over one percent (1.05%).

The second, within-the-mission comparison, however, separates the performances of different detectors. According to this comparison, the GR and BRT methods produced the best overall performance. The detection horizon during the last mission, 2.9 hours, is more than twice as long as the RNN, and nearly twice as long as the GMMs. The FFNN performance, 2.7 hours, was nearly as good. Note that all detectors considerably outperformed thefa existing DTCs, which appeared only 0.1 hour before the failure.

Table 3 lists detection horizons within the last mission with the times required to train the associated detectors. There is little correlation between training time and detection performance; the models with the best detection performance take the longest and shortest times to train. FFNN took by far the most time to train, but it also resulted in the most compact model, which is has an efficient execution and is less prone to overfitting.

Table 3. Performance of algorithms.

Method	Details	Detection Horizon (hours)	Training Time (s)
GMM	$-20 < T < 120$ training; no GMM rejection filtering	0	45
RNN	10+3+10 topology, 8 steps	1.1	670
GMM	$-20 < T < 120$ training; GMM rejection filtering	1.6	45
GMM	$90 < T < 120$ training; GMM rejection filtering	1.6	45
FFNN	3+3 topology	2.7	1780
BRT	$-20 < T < 120$	2.9	40
GR	$-20 < T < 120$	2.9	1

5. DISCUSSION AND CONCLUSIONS

This paper proposes an approach for incremental introduction of PHM capabilities by development of anomaly detection,

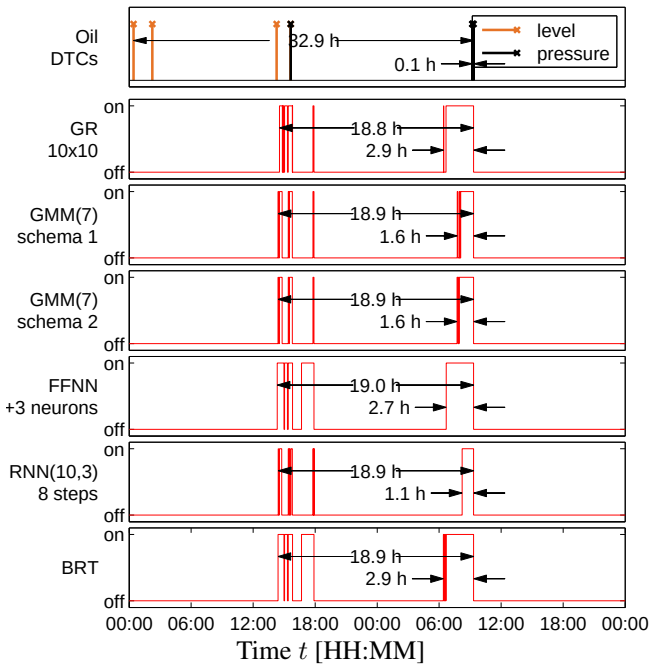


Figure 11. Anomaly detection performance of all models. Each graph shows the on/off state of the anomaly detection using a comparator on the averaged metric. In addition, the top graph shows the diagnostic trouble codes from the vehicle's electronic control unit. For GR see section 3.1.1; for GMM(7)-both schema-see section 3.2.1; for FFNN see section 3.3.1; for RNN see section 3.4.1, and for BRT see section 3.5.1.

even in the presence of a single known failure. We evaluated detectors by disallowing any false alarms during the period of normal operation and measuring detection horizon. The conservative requirement of zero-false-alarm tolerance aimed to compensate for potential overfit problems due to the lack of test and verification data. Rather than waiting to observe a statistically significant set of failures, we propose to start learning from the very first failure instance and carefully consider newly triggered anomalies by verifying the presence of real (incipient) failures. Any new undetected failures would also have to be incorporated in the models. All observed failures and their modes would be documented to allow for future classification and diagnostics, and any observed failure progression, with known failure modes, would be used for future prognostics development. In the context of this vision of PHM, we described its first layer – a tentative anomaly detector that consisted of a pre-filtering, data-driven model, a filter and a threshold comparator. The most space is given to comparison of five candidate data-driven models.

We found that residual based models (GRs, FFNNs, and BRTs) outperformed distance based models (GMMs and RNNs) in this application. The better performance of residual models is probably due to small engineering knowledge that was captured in them by expressing engine oil pressure in terms of engine speed and engine oil temperature. The distance-based models met the nearly zero expert knowledge goal, at least with respect to expert engineering knowledge of the vehicle, but required skills and effort in the machine learning area to select useful models and metrics. In particular, such knowledge and effort was necessary to identify and correct the root cause of the inconsistent GMMs' performance. We reported that locally-optimal GMMs often failed to detect anomalies. To overcome the problem we proposed a heuristic filter that rejects candidate GMMs with non-discriminative components and controls the volume of the largest mixture component.

The old technique of gridded residual, often neglected in favor of more recent methods, not only achieved the best detection horizon, but also trained the fastest. BRT shared the first prize with GR with respect to detection horizon and trained reasonably fast (still not nearly as fast as GR), but its model complexity was much higher. FFNN, by contrast, required by far the most amount of time for training, but achieved a very good result with a the most compact model, which is less likely to overfit. All models performed markedly better than the tradition, vehicle built-in DTCs.

This study employed a very simple anomaly detector – a filter with a threshold comparator. As more failures are observed, more sophisticated inference engine should be considered, especially those that combine multiple learners, such as a model ensemble, which may have a built-in bias against potentially overfitted models.

While this work investigated models for anomaly detection, the results suggest further work to create diagnostic and prognostic algorithms based on these techniques. Implementation of fleet-wide data collection and analysis would allow a statistically significant set of known failures to be created. This in turn would allow estimation of a Receiver Operating Characteristic curve and enable known PHM engineering techniques that are based on such curves to be applied.

ACKNOWLEDGEMENTS

This work was made possible by the Department of the Navy, Office of Naval Research under Award No. N0004-07-1-0823. We gratefully acknowledge our industrial partner, Vnomics Inc. for providing access to the vehicle data.

NOMENCLATURE

Symbol	Definition
ω	Engine speed in radian/s (1 radian/s $\approx$ 9.55 RPM)
ω_i	Sequence of engine speeds in i^{th} speed group
$\Delta\omega$	Width of each speed group
T	Engine oil temperature in °K (°K $\approx$ °C+273)
T_j	Sequence of engine oil temperatures in j^{th} temperature group
ΔT	Width of each temperature group
p	Engine oil pressure in kilo-Pascals (kPa)
p_{ij}	Sequence of pressures in bin of (ω_i, T_j)
p_{ij}^k	k^{th} value of p_{ij}
$\overline{p}_{ij}$	Mean of pressures in bin of (ω_i, T_j)
$\hat{p}$	Estimate of engine oil pressure
S_i	The i^{th} signal
$D_{training}$	Sequence of observed signals used for training
D_{event}	Sequence of observed signals in known event(s)
D_{normal}	Sequence of observed signals during normal operation
$\mathcal{M}$	A model for a set of signals, based on data from a training interval
m	A real-number sequence resulting from evaluating a model over data from a given interval
Θ	Anomaly detection threshold
M_{ij}	The number of values in p_{ij}
ϵ	Residual, or prediction error
z	z-score of prediction error ϵ
σ_p	standard deviation of sequence of prediction errors
$N(\mu, \sigma^2)$	Univariate normal (Gaussian) distribution for mean μ and variance σ^2
$\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal (Gaussian) distribution for mean and covariance $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$
π_k	Weight of distribution $\boldsymbol{N}_k$ in Gaussian Mixture Model
$\mathcal{P}$	Sequence of maximum weighted probabilities signals from Gaussian Mixture Model
pr_k	Probability of (T, ω, p) for k^{th} distribution in Gaussian Mixture Model
l	Posterior likelihood of signal for given model
$\boldsymbol{w}$	Weights in Neural Network
$\sigma(x)$	Logistic sigmoid function of x
OF_t	Outlier Factor for signal at time t

REFERENCES

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Cheifetz, N., Same, A., Aknin, P., & de Verdalle, E. (2011). A pattern recognition approach for anomaly detection on buses brake system. In *Intelligent transportation systems (itsc), 2011 14th international ieee conference on* (pp. 266–271).

Duda, R., Hart, P., & Stork, D. (2000). *Pattern classification. 2nd edn wiley*.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, *77*(4).

Figueiredo, M., & Jain, A. (2002, march). Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(3), 381 -396. doi: 10.1109/34.990138

Golosinski, T. S., Hu, H., & Elias, R. (2001). Data mining vims data for information on truck condition. *Computer Applications in the Minerals Industries*, *88*, 397–402.

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier detection using replicator neural networks. In Y. Kambayashi, W. Winiwarter, & M. Arikawa (Eds.), *Data warehousing and knowledge discovery* (Vol. 2454, p. 113-123). Heidelberg: Springer.

Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., ... others (2004). Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proceedings of siam international conference on data mining* (Vol. 334).

McArthur, S. D. J., Booth, C. D., McDonald, J. R., & McFadyen, I. T. (2005). An agent-based anomaly detection architecture for condition monitoring. *Power Systems, IEEE Transactions on*, *20*(4), 1675-1682.

Nowlan, F., & Heap, H. (1978). *Reliability-centered maintenance* (Tech. Rep. No. AD/A066 579). United Airlines, San Francisco, CA.

Sikorska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*(5), 1803-1836.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Hoboken, NJ: John Wiley & Sons, Inc.

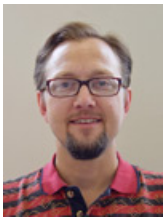
Vnomics. (2012). *Products - vehicle health management software* (Vol. 2012) (No. 3/16/2012).

BIOGRAPHIES

Howard Bussey earned Bachelors degrees in Electrical Engineering, Mathematics, and Psychology from the Universities of Colorado and Minnesota (Boulder, CO, USA and Minneapolis, MN, USA), and a Masters degree in electrical engineering from the University of California (Berkeley, CA, USA). He is CTO/co-founder of Tixlers' Letters, providing educational resources. He was a Senior Staff Engineer in the Golisano Institute of Sustainability, and has also worked at Eastman Kodak, Bell Communications Research, Bell Laboratories, and the National Oceanographic and Atmospheric Administration. He has six patents and 13 publications. He is a member of Eta Kappa Nu and Tau Beta Pi.

Nenad Nenadic received his B.S. in Electrical Engineering from University of Novi Sad (Novi Sad, Serbia) in 1996 and his MS and Ph.D. in Electrical and Computer Engineering from University of Rochester (Rochester, NY, USA) in 1998 and 2001, respectively. He joined Kionix Inc. in 2001, where he worked on development of microelectromechanical inertial sensors. Since 2005, he has been with Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology, where he is currently a Research Associate Professor. His research interest include design, analysis, and monitoring of electromechanical devices and systems. He has two patents in electromechanical design and six publications. He co-authored a textbook "Electromechanics and MEMS". He is a member of IEEE.

Paul Ardis received his B.S. in Computer Science from Purdue University (West-Lafayette, IN, USA) in 2005 and his M.S. and Ph.D. in Computer Science from the University of Rochester (Rochester, NY,USA) in 2007 and 2009 respectively. He is a Lead Scientist at GE Global Research, and was formerly a Research Associate Professor at the Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology and Research Scientist under contract to the Air Force Research Laboratory. He holds two patents in signal processing and has published eleven scholarly articles in machine learning,computer and human vision, biometrics, and machine diagnostics and prognostics. His research interests include data-driven predictive modeling, decision theory, surveillance, natural language processing, and machine sensing.



Michael Thurston received his B.S. and M.S. in Mechanical Engineering from Rochester Institute of Technology (Rochester, NY, USA) in 1988, and his Ph.D. in Mechanical and Aerospace Engineering from the University of Buffalo (Buffalo, NY, USA) in 1998. He is the Technical Director and Research Associated Professor at the Center of Integrated Manufacturing Studies at Rochester Institute of Technology. He formerly held positions in air conditioning system development at General Motor and Delphi, and as a Researcher at the Applied Research Laboratory at Penn State University. He holds 7 patents in the areas of air conditioning and asset health monitoring. His research interests include: sustainable design and production, condition based maintenance and prognostics, and asset health management. He is a member of the Society of Automotive Engineers, and was awarded the Boss Kettering Award for product innovation by Delphi.

Decentralized Approach for Fault Diagnosis of Three Cell Converters

Hanane Louajri, Moamar Sayed-Mouchaweh

Université Lille Nord de France, Mines-Douai, IA 941, rue Charles Bourseul 59508 Douai, France

(e-mail: [hanane.louajri, moamar.sayed-mouchaweh]@mines-douai.fr).

ABSTRACT

In this paper, an approach for fault diagnosis of hybrid dynamic systems (HDS), in particular discretely controlled continuous system, is proposed. The goal is to construct a decentralized diagnosis structure, able to diagnose parametric and discrete faults. This approach considers the system as composed of a set of interacted hybrid components (HCs). Each HC is composed of a discrete component (Dc), e.g. on/off switches, with the continuous components (Ccs), e.g. capacitors, whose continuous dynamic behavior is influenced by the Dc discrete states. A local hybrid diagnosis module, called diagnoser, is associated to each HC in order to diagnose the faults occurring in this HC. In order to take into account the interactions between the different HCs, local diagnosis decisions are merged using a coordinator. The latter issues a final decision about the origin of the fault and identifies its parameters. The advantage of the proposed approach is that local hybrid diagnosers as well as the coordinator are built using local models. The proposed approach is applied to achieve the decentralized diagnosis of discrete and parametric faults of power electronic three-cell converters.

1. INTRODUCTION

1.1 Basic definitions and motivation

A fault can be defined as a non-permitted deviation of at least one characteristic property of a system or one of its components from its normal or intended behavior. Fault diagnosis is the operation of detecting faults and determining possible candidates that explain their occurrence. Most of real systems are hybrid dynamic systems (HDS) (Zaytoon, 2001), (Arogeti *et al.*, 2010) in which the discrete and continuous dynamics cohabit. Therefore, fault diagnosis of HDS must deal with the evolution of continuous dynamics in each discrete mode in order to construct a diagnosis module (called diagnoser) able to diagnose parametric and discrete faults. Parametric faults affect the system continuous dynamics and are characterized by abnormal changes in some system parameters; whereas discrete faults affect the system discrete dynamics and are considered either as the occurrence of unobservable events and/or reaching discrete fault modes. In both cases, they entail unpredicted, abnormal, change in the system configuration. Therefore faults may be modelled in HDS by introducing parameters into the system model, explicit fault events or/and fault modes.

Discretely controlled continuous systems (DCCS) (Schild and Lunze, 2008) are a special class of HDS widely used in the literature. In these systems, the changes in discrete modes are achieved by discrete control commands, e.g. opening or closing a switch.

1.2 State of the art

Many approaches have been proposed in the literature for fault diagnosis of DCCS. They are generally divided into three main categories:

- approaches for the diagnosis of parametric faults,
- approaches for the diagnosis of discrete faults,
- approaches for the diagnosis of both parametric and discrete faults.

In parametric fault diagnosis approaches, (Cocquempot *et al.*, 2004), (Alavi *et al.*, 2011), (Kamel *et al.*, 2012) relations over observable variables are computed in order to generate residuals sensitive to a certain subset of parametric faults in each observable discrete mode.

The discrete fault diagnosis approaches are divided into three main groups. In the first group (Rahiminejad *et al.*, 2012), (Defoort *et al.*, 2011), residuals sensitive to the continuous dynamics in each discrete mode are defined. If unpredicted change occurs due to the occurrence of unobservable discrete fault, the residuals, defined for the

Hanane Louajr et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

discrete mode before the fault occurrence, will be different of zero in the discrete mode after the fault occurrence. This change of residuals values from zero indicates the occurrence of a discrete fault. The approaches of second group (Bhowal *et al*., 2007), (Biswas *et al*., 2006), describe in each normal or fault discrete mode, continuous dynamics as the rate of changes of continuous variables. These rates are considered to be constant. Transition guards are defined as linear inequalities based on continuous variables values. When a guard is satisfied, its corresponding mode transition is enabled. The occurrence of a fault is diagnosed by determining the discrete state reached due to specific guard satisfaction. In the methods of last group (Bayoudh *et al*., 2006), a set of residuals is defined in each normal or fault discrete mode. Each residual is characterized by three symbols: 0, 1 or *und* when the residual value is, respectively, zero, different of zero and undefined. *und* represents the case where the associated residual is not defined in the new active mode. These symbols are used to distinguish the different normal and fault discrete modes. A discrete fault is isolated by determining the current discrete fault mode of the system.

The third category includes few approaches for the diagnosis of both parametric and discrete faults. Some approaches of this category (Derbel *et al*., 2009), capture the continuous dynamics by integrating the occurrence time of events. They consider that the occurrence of discrete or parametric faults does not change events ordering but only alters their timing characteristics. Therefore, a discrete or parametric fault is diagnosed when predicted events occur too late or too early or they do not occur at all during their predefined time intervals. Other methods (Daigle *et al*., 2010), construct temporal causal graphs (TCG) for each normal and fault discrete mode based on the use of a global hybrid bond graph. When measurement deviations, caused by fault occurrence, are observed through residuals, TCG are used to determine the effects that faults will have on the measurements as well as the temporal order in which they deviate. Then, fault signature is defined for each fault as the qualitative value of the magnitude and the first non-zero derivative change which can be observed in the residuals. In order to distinguish parametric from discrete faults, the signatures are extended by adding discrete symbols indicating abrupt changes from zero to non-zero or from non-zero to zero. In (Louajri *et al*., 2013), an approach based on a diagnoser with hybrid structure is developed. It consists of three parts: the discrete diagnoser, the continuous diagnoser and the coordinator. The discrete diagnoser is built using a discrete time hybrid automata representing global model. It exploits the information extracted from the system continuous dynamics to get rid of diagnosis ambiguity due to the system behavior abstraction. The continuous diagnoser generates residuals. The latter compare the measured and nominal values of each continuous variable in order to diagnose the parametric faults in each discrete mode. The information about the discrete mode is provided to the continuous diagnoser thanks to the information extracted from the discrete dynamics. Finally, the coordinator uses the decisions issued from the discrete and continuous diagnosers in order to diagnose faults requiring the interaction between both diagnosers.

1.3 Our approach

Fault diagnosis approaches of the literature do not scale to HDS with a large number of discrete modes because they achieve fault diagnosis using one centralized diagnosis module. The latter is built using a global model of the system. Two problems are arisen -) the weak robustness in the sense that, when the global diagnosis module fails, this may bring down the entire diagnosis task and -) the system global model can be too huge to be physically constructed. Therefore in this paper, the proposed approach of (Louajri *et al*., 2013) is developed to achieve the diagnosis of parametric and discrete faults in decentralized manner using several local hybrid diagnosers. The latter are constructed without the use of a global model of the system but only the local models of the system discrete components (Figure 1).

The paper is organized as follows. In section 2, the three cell converter system is described and modelled. Section 3 defines the steps of the hybrid diagnosis construction. In section 4, a simulation for the three-cell converter is used to demonstrate the efficacy of the approach. A conclusion with the future work ends the paper in section 5.

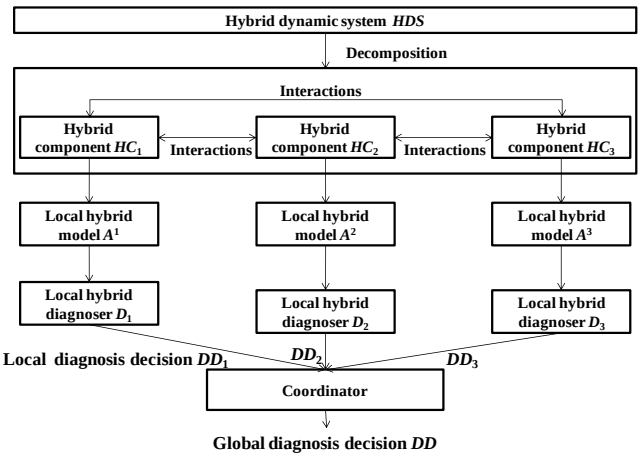


Figure 1. Decentralized hybrid diagnosis structure for a HDS composed of 3 interacted HCs.

2. THREE CELL CONVERTER DESCRIPTION AND MODELING

2.1. System description

In order to illustrate the proposed approach, the decentralized fault diagnosis of three-cell converters

(Shahbazi *et al*., 2013) (Trigeassou, 2011), depicted in Fig.2, is achieved. With the same observability used in the literature (Defoort *et al*., 2011), (Uzunova *et al*., 2012) for the three-cell converter diagnosis, the proposed approach has the advantage to diagnose (not only detect) discrete and parametric faults using a decentralized structure.

The continuous dynamics of the system are described by state vector $X = [Vc_1 \quad Vc_2 \quad I]^T$, where Vc_1 and Vc_2 represent, respectively, the floating voltage of capacitors C_1 and C_2 and I represents the current flowing from source E towards load (R,L) through three elementary switching cells $S_j, j \in \{1,2,3\}$. The latter represent the system discrete dynamics. Each discrete switch S_j has two discrete states: S_j opened ($h_q^j = 0$) or S_j closed ($h_q^j = 1$), where h_q^j is the state discrete output of S_j. The control of this system has two main tasks: -) balancing the voltages between the switches and -) regulating the load current to a desired value. To accomplish that, the controller changes the switches' states from opened to closed or from closed to opened by applying discrete commands 'close' or 'open' to each discrete switch $S_j, j \in \{1,2,3\}$ (see Fig.2). Thus, the considered example is a DCCS.

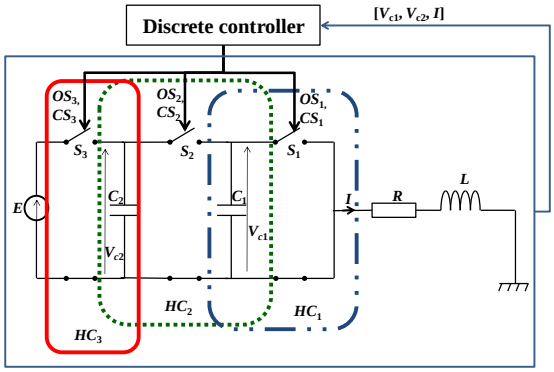


Figure 2. Three-cell converter discription and decomposition

2.2. System modeling and decomposition

The real system dynamic evolution of three-cell converter is written as (Defoort *et al*., 2011)

$$\begin{cases} \dot{V}c_1 = -h_q^1 \frac{1}{C_1} I + h_q^2 \frac{1}{C_1} I \\ \dot{V}c_2 = -h_q^2 \frac{1}{C_2} I + h_q^3 \frac{1}{C_2} I \\ \dot{I} = -\frac{R}{L} I + h_q^1 \frac{1}{L} Vc_1 + h_q^2 \frac{1}{L} (Vc_2 - Vc_1) + h_q^3 \frac{1}{L} (E - Vc_2) \end{cases} \quad (1)$$

As shown in (1), the discrete state of S_1, represented by a real discrete output h_q^1, influences the dynamic evolution of Vc_1 and I. The discrete state of S_2, represented by h_q^2, impacts the dynamic evolution of Vc_1, Vc_2 and I. The discrete state of S_3, represented by h_q^3, influences the dynamic evolution of Vc_1 and Vc_2. Thus, the three-cell converter system is decomposed into three interacted HCs as shown in Fig.2:

- HC_1 is composed of switch S_1 (Dc_1), Vc_1 (Cc_1) and I (Cc_3).
- HC_2 is composed of switch S_2 (Dc_2), Vc_1 (Cc_1), Vc_2 (Cc_2) and I (Cc_3).
- HC_3 is composed of switch S_3 (Dc_3), Vc_2 (Cc_2) and I (Cc_3).

In the literature (Defoort *et al*., 2011), (Uzunova *et al*., 2012), eight faults are considered for the diagnosis of the three-cell converters system (Table 1).

Table 1. Faults for the diagnosis of three-cell converters

Fault types	*Fault labels*	*Fault description*
Discrete faults	F_1	S_1 stuck opened
	F_2	S_1 stuck closed
	F_3	S_2 stuck opened
	F_4	S_2 stuck closed
	F_5	S_3 stuck opened
	F_6	S_3 stuck closed
Parametric faults	F_7	Change in the nominal parameter values of C_1due to C_1 ageing
	F_8	change in the nominal parameter values of C_2 due to C_2 ageing

Labels N_1, N_2 and N_3 signify the normal operating modes for, respectively, HC_1, HC_2 and HC_3.

2.3. Residuals generation

In order to show the influence of each discrete component on the dynamic evolution of each continuous component, (1) is rewritten as follows:

$$\begin{cases} \dot{V}c_1 = \dot{V}c_1^1 + \dot{V}c_1^2 \\ \dot{V}c_2 = \dot{V}c_2^2 + \dot{V}c_2^3 \\ \dot{I} = \dot{I}_c + \dot{I}^1 + \dot{I}^2 + \dot{I}^3 \end{cases} \quad (2)$$

where $\dot{V}c_1^1 = -h_q^1 \frac{1}{C_1} I$, $\dot{V}c_1^2 = h_q^2 \frac{1}{C_1} I$, $\dot{V}c_2^2 = -h_q^2 \frac{1}{C_2} I$, $\dot{V}c_2^3 = h_q^3 \frac{1}{C_2} I$, $\dot{I}_c = -\frac{R}{L} I$, $\dot{I}^1 = h_q^1 \frac{1}{L} Vc_1$, $\dot{I}^2 = h_q^2 \frac{1}{L} (Vc_2 - Vc_1)$, $\dot{I}^3 = h_q^3 \frac{1}{L} (E - Vc_2)$.

$\dot{V}c_1^1$ represents the real dynamic evolution of Vc_1 according to the discrete state of S_1 (Dc_1). Likewise, $\dot{V}c_1^2$, $\dot{V}c_2^2$, $\dot{V}c_2^3$, $\dot{I}^1$, $\dot{I}^2$and $\dot{I}^3$ have the same definition as $\dot{V}c_1^1$. $\dot{I}_c$ represents the part of dynamic evolution of I which does not depend on the discrete state of any switch.

Similarly, considering that the parametric faults related to the load (R, L) are not considered, the equations system for the nominal dynamic evolution of system components can be written as:

$$\begin{cases} \tilde{V}c_1 = \tilde{V}c_1^1 + \tilde{V}c_1^2 \\ \tilde{V}c_2 = \tilde{V}c_2^2 + \tilde{V}c_2^3 \\ \tilde{I} = \tilde{I}_c + \tilde{I}^1 + \tilde{I}^2 + \tilde{I}^3 \end{cases} \quad (3)$$

$\tilde{V}c_1^1 = -\tilde{h}_q^1 \frac{1}{\tilde{C}_1} I$, $\tilde{V}c_1^2 = \tilde{h}_q^2 \frac{1}{\tilde{C}_1} I$, $\tilde{V}c_2^2 = -\tilde{h}_q^2 \frac{1}{\tilde{C}_2} I$, $\tilde{V}c_2^3 = \tilde{h}_q^3 \frac{1}{\tilde{C}_2} I$, $\tilde{I}_c = \dot{I}_c = -\frac{R}{L} I$ $\tilde{I}^1 = \tilde{h}_q^1 \frac{1}{L} Vc_1$, $\tilde{I}^2 = \tilde{h}_q^2 \frac{1}{L}(Vc_2 - Vc_1)$, $\tilde{I}^3 = \tilde{h}_q^3 \frac{1}{L}(E - Vc_2)$ where $\tilde{h}_q^1$, $\tilde{h}_q^2$ and $\tilde{h}_q^3$ are the nominal values of states q_1, q_2 and q_3 discrete outputs while $\tilde{C}_1$ and $\tilde{C}_2$ are the nominal values of C_1 and C_2. Based on (2) and (3), residuals r_1, r_2 and r_3 are generated as follows:

$$\begin{cases} r_1 = \left(-\tilde{h}_q^1 \frac{1}{\tilde{C}_1} + h_q^1 \frac{1}{C_1}\right) I + \left(\tilde{h}_q^2 \frac{1}{\tilde{C}_1} - h_q^2 \frac{1}{C_1}\right) I \\ r_2 = \left(-\tilde{h}_q^2 \frac{1}{\tilde{C}_2} + h_q^2 \frac{1}{C_2}\right) I + \left(\tilde{h}_q^3 \frac{1}{\tilde{C}_2} - h_q^3 \frac{1}{C_2}\right) I \\ r_3 = \left(\tilde{h}_q^1 - h_q^1\right) \frac{Vc_1}{L} + \left(\tilde{h}_q^2 - h_q^2\right) \frac{(Vc_2 - Vc_1)}{L} \\ \qquad + \left(\tilde{h}_q^3 - h_q^3\right) \frac{(E - Vc_2)}{L} \end{cases} \quad (4)$$

In order to show the influence of each discrete component on the residuals, (4) is rewritten as follows:

$$\begin{cases} r_1 = r_1^1 + r_1^2 \\ r_2 = r_2^2 + r_2^3 \\ r_3 = r_{3c} + r_3^1 + r_3^2 + r_3^3 \end{cases} \quad (5)$$

where $r_1^1 = \left(-\tilde{h}_q^1 \frac{1}{\tilde{C}_1} + h_q^1 \frac{1}{C_1}\right) I = \left(\tilde{V}c_1^1 - \dot{V}c_1^1\right)$, $r_1^2 = \left(\tilde{V}c_1^2 - \dot{V}c_1^2\right)$, $r_2^2 = \left(\tilde{V}c_2^2 - \dot{V}c_2^2\right)$, $r_2^3 = \left(\tilde{V}c_2^3 - \dot{V}c_2^3\right)$, $r_3^1 = \left(\tilde{I}^1 - \dot{I}^1\right)$, $r_3^2 = \left(\tilde{I}^2 - \dot{I}^2\right)$, $r_3^3 = \left(\tilde{I}^3 - \dot{I}^3\right)$ and $r_{3c} = \left(\tilde{I}_c - \dot{I}_c\right) = 0$.

2.4. Hybrid automata construction

Hybrid automata A^1 characterizing the hybrid dynamics of HC_1 is defined by the tuple (see Fig.4 and Fig.4):

$$A^1 = (Q^1, \Sigma^1, SP^1, \delta^1, X^1, flux^1, r^1, Init^1) \quad (6)$$

where,

Q^1={S_1O (S_1 opened), S_1C (S_1 closed), S_1SO (S_1 stuck opened), S_1SC (S_1 stuck closed)}: is a finite set of discrete states (discrete modes) of S_1. The output of state q_k^1 is characterized by real discrete output vector $h_q^1 \in$ {0 (when S_1 is opened), 1(when S_1 is closed)} and nominal discrete output vector $\tilde{h}_q^1 =$ {0 (when S_1 have to be opened), 1(when S_1 have to be closed)}. At normal discrete mode (state) $\tilde{h}_q^1 = h_q^1$ while in faulty mode $\tilde{h}_q^1 \neq h_q^1$;

$\Sigma^1 = \Sigma_o^1 \cup \Sigma_u^1$ is the event set of S_1. It includes observable events corresponding to control command events $\Sigma_o^1 =$ {CS_1(close S_1), OS_1(open S_1)} and unobservable events Σ_u^1 including fault events.$\Sigma_f^1 = \{S_1_stuck_open, S_1_stuck_close, f_{\tilde{C}_1 \neq C_1}\}$ denotes the set of fault events (discrete and parametric) that can occur in HC_1. The set of fault events contains three different fault types or modes indicated by the fault labels: { F_1, F_2, F_7 }. The set of labels for HC_1 is $SP^1 =$ { N_1, F_1, F_2, F_7} .

$\delta^1: Q^1 \times \Sigma^1 \rightarrow Q^1$: is the state transition function. A transition $\delta^1(q^1, e) = q^{1+}$ corresponds to a change from state q^1 to state q^{1+} after the occurrence of event $e \in \Sigma^1$;

$X^1 = \{Vc_1, Vc_2, I\}$ is a finite set of continuous variables associated to S_1;

$flux^1: Q^1 \times X^1 \rightarrow \mathbb{R}^n = \{\tilde{\dot{X}}^1, \dot{X}^1\}$: is a function characterizing temporal evolution $\dot{X}^1$ and nominal evolution $\tilde{\dot{X}}^1$ of continuous variables X^1 in each discrete state q_k^1, where $\tilde{\dot{X}}^1 = \left[\tilde{\dot{V}}c_1^1 \quad \tilde{\dot{V}}c_2^1 \quad \tilde{\dot{I}}^1\right]^T$, $\dot{X}^1 = [\dot{V}c_1^1, \quad \dot{V}c_2^1 \quad \dot{I}^1]^T$;

$Init^1 \subset Q^1 \times X^1 = S_1O\ (q_0^1)$: is the set of initial conditions.

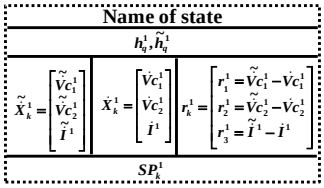


Figure 3. Hybrid state of A^1 for HC_1.

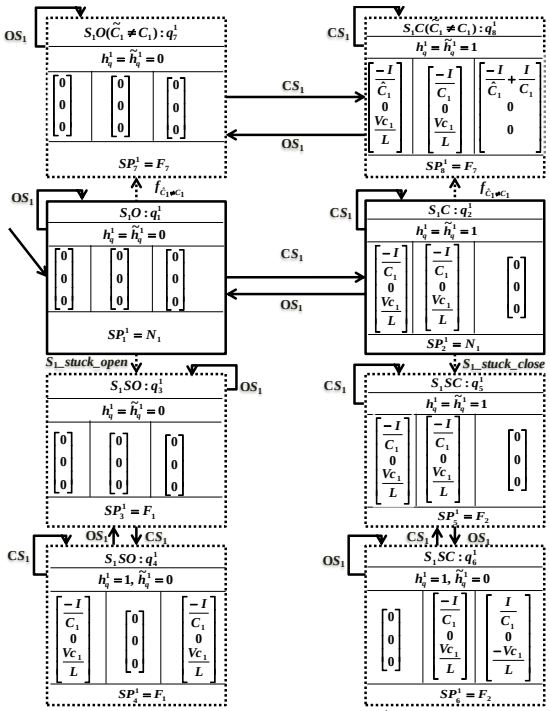


Figure 4. Hybrid automata A^1 for HC_1.

$r^1 = \{r_1^1, r_2^1, r_3^1\}$: is a set of residuals associated to HC_1;

Since Vc_2 does not belong to HC_1, therefore, $\tilde{V}c_2^1 = 0$,

$\dot{V}c_2^1 = 0$. Thus r_2^1 is equal to zero.

Hybrid automata A^2 and A^3 for HC_2 and HC_3 are constructed by the same manner.

2.5. Motivation to use the considered residuals

Let us consider the occurrence of a fault of type F_5, e.g. S_3 stuck opened. When the controller sends control command 'CS_3'(close S_3), S_3 remains in its stuck-opened mode ($\tilde{h}_q^3 = 1$ and $h_q^3 = 0$). The occurrence of a fault of type F_5 impacts at the same time r_2 and r_3 ($\tilde{h}_q^3 = 1$ and $h_q^3 = 0$) while it does not impact r_1 ($(\tilde{h}_q^1 = h_q^1)$, $(\tilde{h}_q^2 = h_q^2)$ and $(\tilde{C}_1 = C_1)$), see (4). Therefore, there is no delay of the influence of the fault occurrence on the sensitive residuals, e.g. r_2 and r_3. Moreover, there is no fault propagation from one residual to another one, from r_2 or r_3 towards r_1.

3. THREE CELL CONVERTER DIAGNOSIS

3.1. Global fault signature construction

A qualitative signature is constructed by generating continuous and discrete symbols from residual values. Continuous symbols $CS(r_i) \in \{0, -, +\}$ represent the qualitative abstraction of residual values into stable/increasing/decreasing ones:

- r_i^0: $r_i(t)$ belongs to the nominal interval;
- r_i^-: $r_i(t)$ is below the nominal interval;
- r_i^+: $r_i(t)$ is above the nominal interval.

The occurrence of a discrete fault exhibits an abrupt change in the continuous dynamics due to unpredicted change in Dc_j discrete mode. This change is characterized by the absence ($h_q^j = 0$ while $\tilde{h}_q^j = 1$) or the addition ($h_q^j = 1$ while $\tilde{h}_q^j = 0$) of associated term e.g., $\frac{I}{C_1}$. On the other hand, parametric faults due to the ageing effect cannot cause this abrupt change with a finite change in magnitude. In fact, they are indicated by a progressive abnormal change of the parameter value. In order to take into account this discriminative information, discrete symbols $DS(r_i)$ are added for the abstraction of each residual r_i in order to distinguish between parametric and discrete faults as follows:

- $PC_i^j = +Val$: denotes an abrupt positive change in residual r_i due to a discrete fault caused by Dc_j. $+Val$ is equal to the absolute value of the term associated to h_q^j;
- $NC_i^j = -Val$: denotes an abrupt negative change in residual r_i due to a discrete fault caused by Dc_j;
- UC_i: denotes that there is no observed abrupt change in residual r_i.
- A fault signature Sig_q at global discrete state q is the combination of continuous and discrete symbols of the different residuals as follows:

$$Sig_q = \left(r_1^{CS(r_1)}, DS(r_1)\right) \& \dots \& \left(r_n^{CS(r_n)}, DS(r_n)\right) \quad (7)$$

3.2. Local fault signature construction

Each discrete state q_k^j of A^j generates a fault signature sig_k^j as a guard over residuals r^j calculated in this discrete state as follows:

$$sig_k^j = \left(r_1^{j\,CS(r_1^j)}, DS(r_1^j)\right) \& \dots \& \left(r_n^{j\,CS(r_n^j)}, DS(r_n^j)\right) \quad (8)$$

Based on (5), we can write:

$r_i = \tilde{x}_i - \dot{x}_i = \left(\tilde{x}_i^1 - \dot{x}_i^1\right) + \dots + \left(\tilde{x}_i^l - \dot{x}_i^l\right) = r_i^1 + \dots + r_i^l$

If $\left[\left(\tilde{x}_i^j - \dot{x}_i^j\right) = r_i^j\right] \neq 0$, it means that the other parts of residual r_i are equal to zero (one fault can be occurred at the same time). In this case, $r_i = r_i^j$. Hence, r_i will have the continuous and discrete symbols of r_i^j. Thus (8) is rewritten as follows:

$$sig_k^j = \left(r_1^{CS(r_1)}, DS(r_1)\right) \& \dots \& \left(r_n^{CS(r_n)}, DS(r_n)\right) \quad (9)$$

By comparing (8) and (9), we can notice that sig_k^j becomes equivalent to the global fault signature Sig_q.

3.3. Local hybrid diagnoser

The objective of local hybrid diagnoser D_j is to detect and isolate the occurrence of parametric and discrete faults affecting the dynamics of hybrid component HC_j. D_j is built based on the local model, A^j, of HC_j. Each state of D_j, denoted z_k^j, is of the form shown in Fig.5.

Figure 5. State of local hybrid diagnoser D_j of HC_j.

Local hybrid diagnoser D_1 of HC_1 is depicted in Fig.6. It is constructed from hybrid automata A^1 of Fig.4.

D_1 is constructed as follows:

- Initial state z_1^1, characterized by (Q_1^1, $\tilde{X}^1$, SP^1), is composed of the following A^1 states: q_1^1 (A^1 initial state), q_3^1 reached from q_1^1 by the occurrence of a fault event '$S_1_stuck_open$' (fault of type F_1) and q_7^1 reached from q_1^1 due to the occurrence of a fault event '$f_{\tilde{C}_1 \neq C_1}$' (fault of type F_7). Thus, Q_1^1 is equal to $\{q_1^1, q_3^1, q_7^1\}$. SP^1 gathers the normal and fault labels associated to the states belonging to Q_1^1. Therefore, SP^1 is equal to $\{N_1, F_1, F_7\}$.

Finally, $\tilde{\tilde{X}}^1$ gathers $\tilde{\tilde{X}}_k^1$ of all the states q_k^1 of Q_1^1. Since states q_3^1 and q_7^1 are reached from q_1^1 due to the occurrence of unobservable event (a fault), $\tilde{\tilde{X}}_1^1$, $\tilde{\tilde{X}}_3^1$ and $\tilde{\tilde{X}}_7^1$ are equivalent and equal to $[0 \quad 0 \quad 0]^T$ (see Fig.4).

- The states of D_1 reached due to the occurrence of each control command event observed by HC_1 are computed. Since D_1 initial state is S_1O, control command OS_1 will not change D_1 state z_1^1. The event CS_1 transits D_1 from z_1^1 to z_2^1, characterized by (Q_2^1, $\tilde{\tilde{X}}^1$, SP^1). Q_2^1 is equal to all the states reached from Q_1^1 due to the occurrence of CS_1. Thus, Q_2^1 is equal to $\{q_2^1, q_4^1, q_8^1\}$ (see Fig.4). Moreover, all the states of A^1 reached from Q_2^1 due to the occurrence of unobservable event are added to Q_2^1. Therefore, Q_2^1 is equal to $\{q_2^1, q_4^1, q_8^1, q_5^1\}$. SP^1 is equal to $\{N_1, F_1, F_7, F_2\}$ while $\hat{\tilde{X}}^1$ is equal to $\left[\frac{-I}{C_1} \quad 0 \quad \frac{Vc_1}{L}\right]^T$ (see Fig.4).
- Fault signatures are generated for each D_1 state thanks to the continuous dynamic evolution in each discrete state of Q_k^1. In the initial D_1 state, z_1^1, the continuous dynamic evolution in any state of Q_1^1 does not evolve. Therefore, their associated residuals are equal to zero leading to obtain the fault signature Sig_N^1 (see Table 2). In z_2^1, the continuous dynamic evolution of the states belonging to Q_2^1 will allow to generate four fault signatures as we can see in Fig.6. They allow to detect and isolate discrete and parametric faults F_1 and F_7 as follows. q_4^1 of A^1 (reached due to the occurrence of fault of type F_1) generates local fault signature $Sig_{q_4^1}^1$.

$Sig_{q_4^1}^1 = \left(r_1^{1-}, \frac{-I}{C_1}\right) \& (r_2^{10}, UC_2) \& \left(r_3^{1+}, \frac{Vc_1}{L}\right)$ (see the values of local residuals in q_4^1 of Fig.4). As explained in subsection 3.2, local fault signature $Sig_{q_4^1}^1$ is equal to global fault signature.

$$Sig_q = \left(r_1^{CS(r_1)}, DS(r_1)\right) \& \left(r_2^{CS(r_2)}, DS(r_2)\right) \& \left(r_3^{CS(r_3)}, DS(r_3)\right) = \left(r_1^-, \frac{-I}{C_1}\right) \& (r_2^0, UC_2) \& \left(r_3^+, \frac{Vc_1}{L}\right).$$

This global signature is used as transition to isolate the occurrence of a fault of type F_1. Same reasoning can be applied for the other fault signatures. To overcome the noise problem, the values of comparison (e.g.,$\frac{I}{C_2}$) are replaced by the intervals corresponding to the selected confidence level. These intervals are calculated using Z-test in order to determine the thresholds of each value.

Same reasoning can be followed for the construction of the other states of D_1.

It is worth pointing out that $Sig_q = Sig_{q_4^1}^1 = \left(r_1^-, \frac{-I}{C_1}\right) \& (r_2^0, UC_2) \& \left(r_3^+, \frac{Vc_1}{L}\right)$ means that the three conditions have to be satisfied in order to enable the corresponding transition.

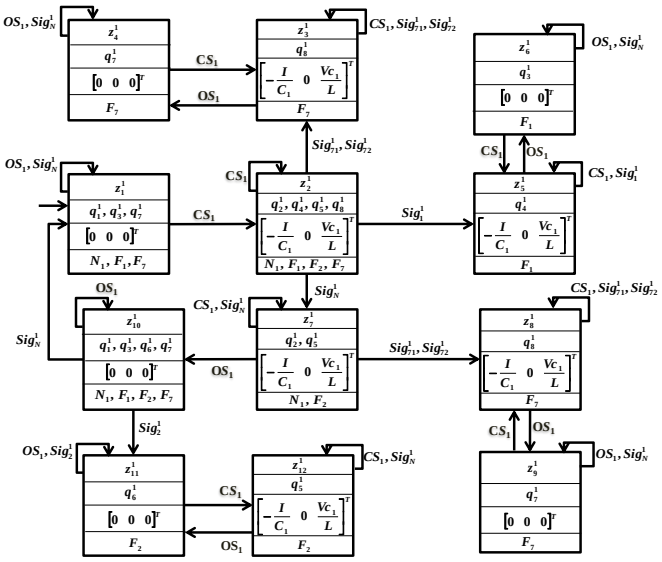


Figure 6. Local hybrid diagnoser D_1 of HC_1.

Table II shows the local fault signatures (equivalent to the global fault signatures) used by D_1 to achieve its local diagnosis.

Table 2. Local fault signatures generated due to the occurrence of faults in HC_1.

SP^1	*Local signature name*	*Equivalent global fault signatures*
F_1	Sig_1^1	$\left(r_1^-, \frac{-I}{C_1}\right) \& (r_2^0, UC_2) \& \left(r_3^+, \frac{Vc_1}{L}\right)$
F_2	Sig_2^1	$\left(r_1^+, \frac{I}{C_1}\right) \& (r_2^0, UC_2) \& \left(r_3^-, \frac{-Vc_1}{L}\right)$
F_7	Sig_{71}^1	$(r_1^-, UC_1) \& (r_2^0, UC_2) \& (r_3^0, UC_3)$
	Sig_{72}^1	$(r_1^+, UC_1) \& (r_2^0, UC_2) \& (r_3^0, UC_3)$
N_1	Sig_N^1	$(r_1^0, UC_1) \& (r_2^0, UC_2) \& (r_3^0, UC_3)$

The other diagnosers D_2 and D_3 for HC_2 and HC_3 can be constructed similarly as for D_1. D_2 is sensitive to discrete faults F_3 and F_4 and to parametric faults F_7 and F_8, while D_3 is sensitive to discrete faults F_5 and F_6 and to parametric fault F_8. The occurrence of parametric fault F_7 (respectively F_8) is detected intrinsically by D_1 and D_2 (respectively D_2 and D_3).

3.4. Coordinator construction

The system decomposition achieved by the proposed approach allows each local hybrid diagnoser to diagnose faults that can occur in its corresponding hybrid component. In order to obtain a decentralized diagnosis performance equivalent to a centralized diagnoser, a decision coordinator is defined. It generates a global diagnosis decision by merging local diagnosis decisions provided by local hybrid diagnosers. Let us denoted F^1, F^2 and F^3 the faults that can

occur, respectively, in HC_1, HC_2 and HC_3. $F^1 \in \{F_1, F_2, F_7\}$, $F^2 \in \{F_3, F_4, F_7, F_8\}$ and $F^3 \in \{F_5, F_6, F_8\}$,. Global diagnosis decision DD is computed as follows:

- D_1 diagnoses with certainty the occurrence of a fault of type F^1 through the global fault signature Sig_q. D_2 cannot diagnose with certainty the occurrence of this fault because it does not belong to its associated HC_2. D_3 cannot diagnose with certainty the occurrence of this fault because it does not belong to its associated HC_3. Therefore, the global diagnosis decision will be $DD = F^1$.
- Global fault signature Sig_q corresponds to a fault of type F^1 or of type F^2 (F_7). Thus, global diagnoser DD will be F^1 or F^2. Both D_1 and D_2 are sensitive to this fault signature, therefore D_1 declares F_7; and D_2 declares F_7. In order to obtain a decentralized diagnosis decision equivalent to the global one, global diagnosis decision DD will be equal to (F^1 or F^2) $= F_7$.
- Table 3 shows global diagnosis decision DD. A local diagnoser declares 'nothing' when it cannot confirm the occurrence or the non-occurrence of a fault.

Table 3. Global diagnosis decision DD for Three Cell Converter .

cases	*Local diagnoser D_1*	*Local diagnoser D_2*	*Local diagnoser D_3*	*Global decision DD*
1	N_1	N_2	N_3	N
2	F^1	N_2 or Nothing	N_2 or Nothing	F^1
3	N_1 or Nothing	F^2	N_3 or Nothing	F^2
4	F^1	F^2	N_3 or Nothing	F^1or F^2
5	N_1 or Nothing	F^2	F^3	F^2or F^3
6	N_1 or Nothing	N_2 or Nothing	F^3	F^3
7	Nothing	Nothing	Nothing	Nothing

3.5. Identification of parametric faults

When one of parametric faults is diagnosed, its real value needs to be identified. As an example, for parametric fault of type F_7 related to C_1, the real value of the latter is identified based on its corresponding residual as follows:

$$r_1 = \left(-\frac{1}{\tilde{C}_1} + \frac{1}{C_1}\right) \Rightarrow C_1 = \frac{\tilde{C}_1}{\tilde{C}_1 r_1 + 1} \qquad (10)$$

$$r_2 = \left(+\frac{1}{\tilde{C}_1} - \frac{1}{C_1}\right) \Rightarrow C_1 = \frac{\tilde{C}_1}{\tilde{C}_1 r_2 - 1} \qquad (11)$$

The same reasoning is applied to identify the real value of capacitor C_2 in case of fault of type F_8 related to C_2.

4. EXPERIMENTATION AND OBTAINED RESULTS

In order to evaluate the proposed approach, simulations were carried out for the three-cell converter using Matlab-Simulink™ environment and Stateflow™ toolbox. The parameters used in these simulations are:

$E = 60V, \tilde{C}_1 = \tilde{C}_2 = 40\mu F, R = 200\Omega, L = 0.1H.$

In order to highlight the efficiency of the diagnoser, the simulations take into account the set of faults defined in Table 1 for the three-cell converter.

Discrete controller commands are assured by a pulse width modulation (PWM) signal (Defoort *et al.*, 2011). Fig.7 depicts the control of three switches S_1,S_2 and S_3. When the triangular signal is below the reference signal (ref in Fig.7), the associated switch is controlled to be opened. When the triangular signal is above the reference signal, the associated switch is controlled to be closed. This sequence of control is periodic with a period of $T_{PWM} = 0.02\ s$.

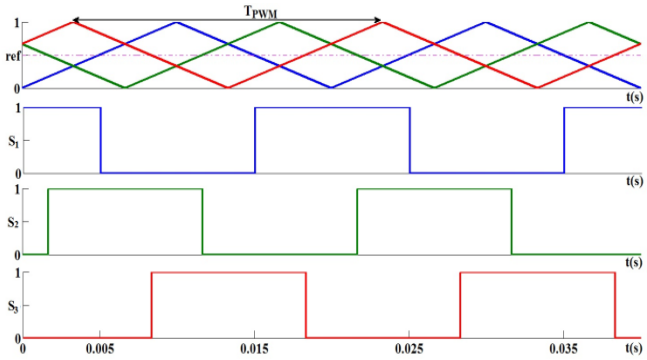


Figure 7. PWM for control of three switches S_1, $S_2$and S_3.

4.1. Normal conditions scenario

Fig.8 depicts, respectively, the signals of floating voltages Vc_1 and Vc_2 and the current I. These signals correspond to the normal conditions. Moreover, one can see in Fig.8 that Vc_1 (respectively Vc_2) has a periodic signal corresponding to load and unload of capacitor C_1 (respectively C_2) around the mean value $Vc_{1ref} = \frac{E}{3} = 20V$ (respectively$Vc_{2ref} = \frac{2E}{3} = 40V$) and that the current I remains constant in the region of its reference value (0.15A).

Fig.9 shows the real and nominal dynamic evolution of Vc_1 ($\dot{Vc}_1$and $\tilde{\dot{V}}c_1$), Vc_2 ($\dot{Vc}_2$and $\tilde{\dot{V}}c_2$) and I($\dot{I}$and $\tilde{\dot{I}}$). We can notice that the curves representing the real and nominal dynamic evolutions are superposed. Consequently, residuals r_1, r_2 and r_3 are equal to zero in these conditions.

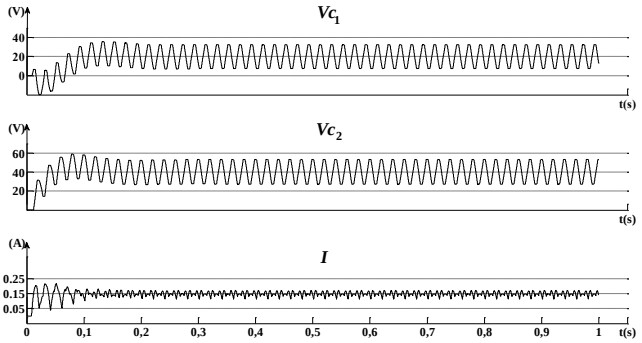


Figure 8. Real signals corresponding to Vc_1, Vc_2 and I in normal conditions .

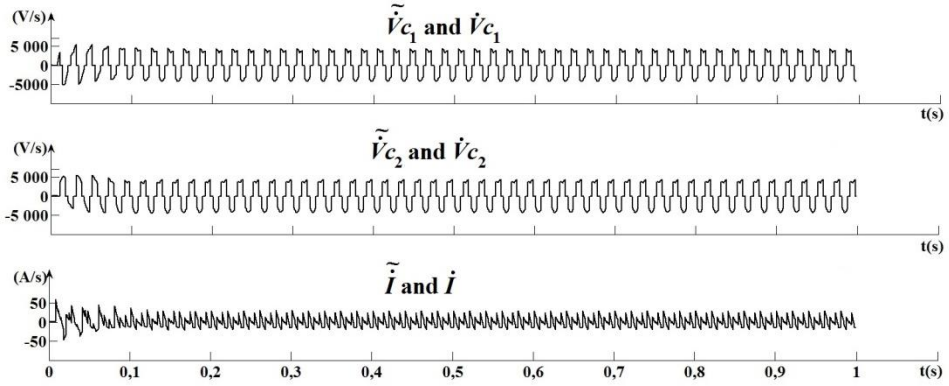


Figure 9. Real and nominal dynamic evolution of Vc_1, Vc_2 and I in normal conditions.

4.2. Faulty conditions scenario

The test scenario is generated as follows (see Fig.10). Each fault *f*, belonging to one of the fault labels of Table 1, is generated starting at time t_{sf} and ending at time t_{ef}. Then, the system returns to normal operating conditions before generating a new fault for a certain time. Parametric faults of types F_7 and F_8 are simulated by changing gradualy the real values of C_1, respectively C_2, in positive or negative direction using a ramp signal. Vc_1 , Vc_2 and I simulated signals including these faults are represented in Fig. 11.

One can see in Fig.11 that Vc_1 (respectively Vc_2) has lost the periodic aspects in the case of fault and that the current I has become nonconstant in the region of its reference value.

r_1, r_2, r_3 are represented in Fig.12 and Fig.13. As expected, r_1 is sensitive to the faults of types F_1, F_2, F_3, F_4 and F_7, $r_2$is sensitive to the faults of types F_3, F_4, F_5, F_6 and F_8 while r_3 is sensitive to the faults of types F_1, F_2, F_3, F_4, F_5 and F_6.

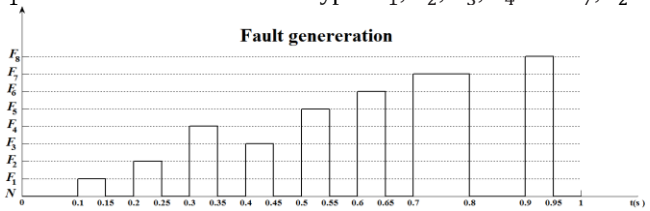


Figure 10. Time of apperance, injection, of faults during the simulation of three cell converter.

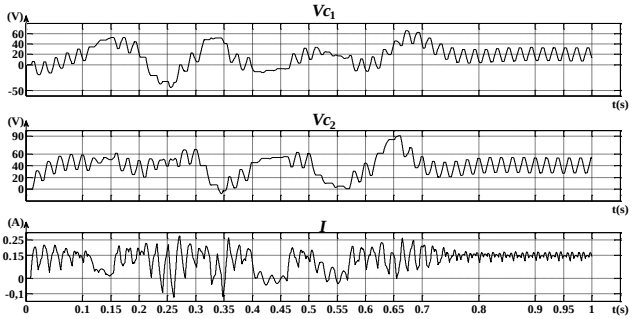


Figure 11. Real signals of Vc_1, Vc_2 and I in faulty and normal conditions.

Fig.14, Fig.15, Fig.16 and Fig.17 show, respectively, local decision (SP_1) of diagnoser D_1, local decision (SP_2) of diagnoser D_2, local decision (SP_3) of diagnoser D_3 and global decision (SP).

The first local diagnoser D_1 is sensitive to faults of types F_1, F_2 and F_7 (diagnosis with certainty their ocurence), the second local diagnoser D_2 is sensitive to faults of types F_3, F_4, F_7 and F_8 while the third local diagnoser D_3 is sensitive to faults of types F_5, F_6 and F_8. We can conclude that the global decision indicates with certainty the occurrence of each of the generated faults. The diagnosis delay corresponds to the time when the system is in a discrete fault is due to residues that are silent in some discrete state.

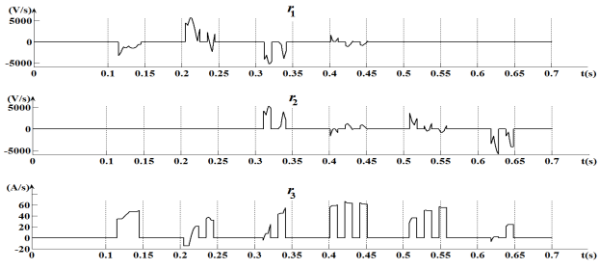


Figure 12. Residuals corresponding to generated discrete faults of Fig.10.

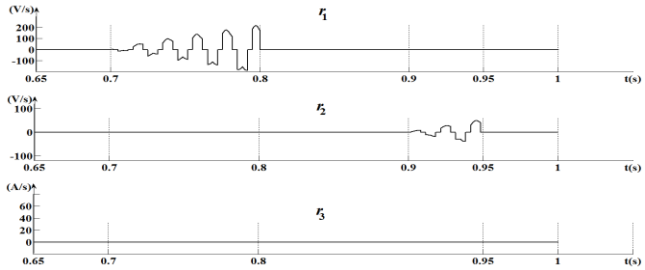


Figure 13. Residuals corresponding to generated parametric faults of Fig.10.

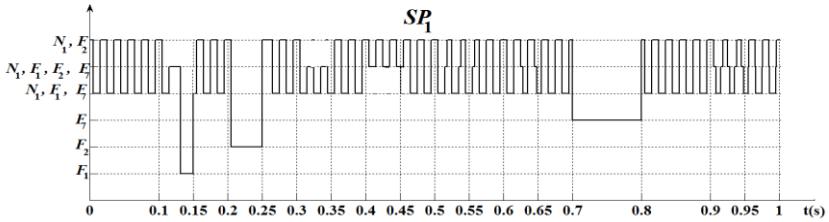


Figure 14. Local decision (SP_1) of D_1.

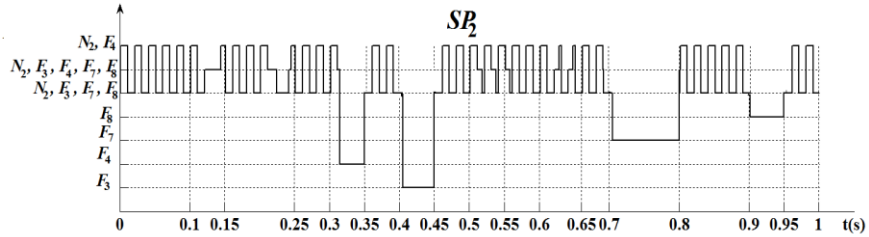


Figure 15. Local decision (SP_2) of D_2.

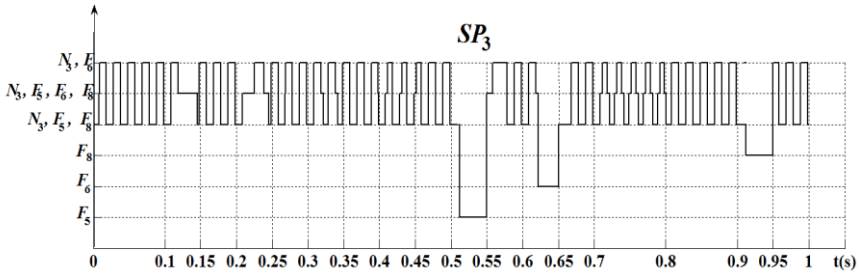


Figure 16. Local decisions (SP_3) of D_3.

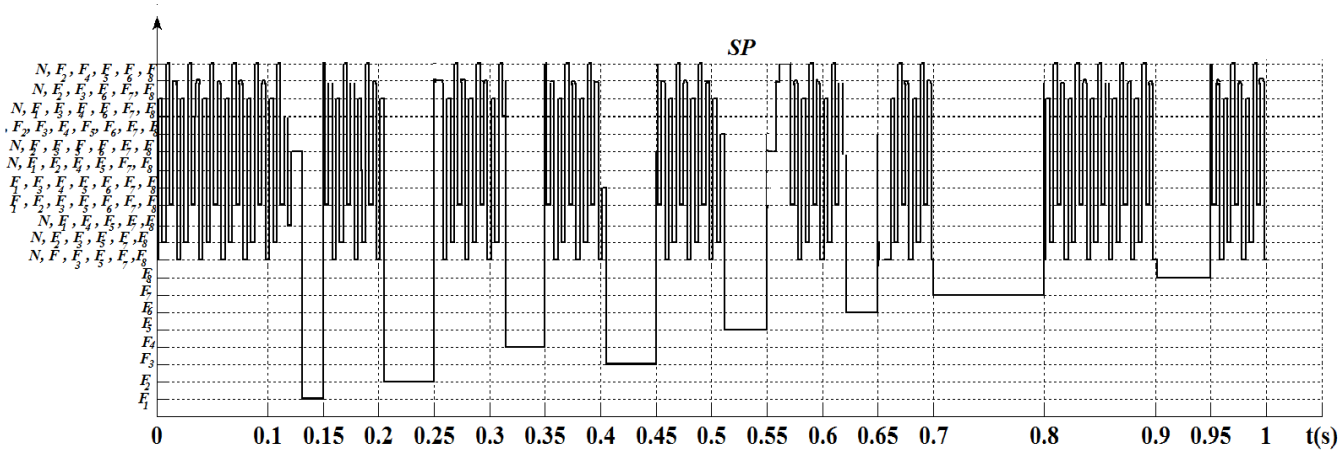


Figure 17. Global diagnosis decision issued by the coordinator.

4.3. Normal conditions with noises in parameters scenario

Diagnosis algorithms should be tested and evaluated on real systems with practical significance. In these systems, factors such as noise make diagnosis challenging. Therefore, there is a need to evaluate the robustness of the diagnosis algorithms for different fault and noise magnitudes. Accurate simulation models of the system are required for this purpose. Further, it is important to execute the diagnosis algorithms on systems, where model uncertainty is always present, and complicates the diagnosis task. In order to examine the robustness of our approach, a parametric noise (see for example Fig.18), applied on parameters, is used. From an electrical point of view, the resistors are the most disturbing element in tree cell converter systems. For this

reason, we simulated noise on signal resistance.

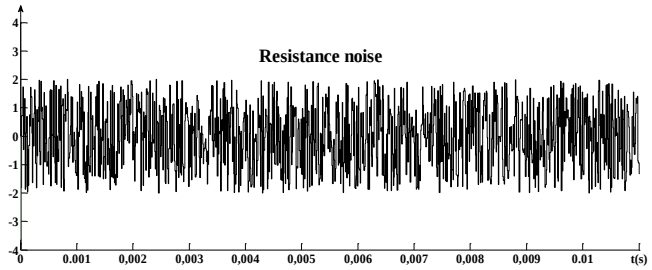


Figure 18. Noise added to resistance R in the converter.

In order to take into account the noises in R, the residuals of (4) is written as follows:

$$\begin{cases} r_1 = \left(-\tilde{h}_q^1 \frac{1}{\tilde{C}_1} + h_q^1 \frac{1}{C_1}\right) I + \left(\tilde{h}_q^2 \frac{1}{\tilde{C}_1} - h_q^2 \frac{1}{C_1}\right) I \\ r_2 = \left(-\tilde{h}_q^2 \frac{1}{\tilde{C}_2} + h_q^2 \frac{1}{C_2}\right) I + \left(\tilde{h}_q^3 \frac{1}{\tilde{C}_2} - h_q^3 \frac{1}{C_2}\right) I \\ r_3 = \left(-\tilde{R} + R_b\right) \frac{I}{L} + \left(\tilde{h}_q^1 - h_q^1\right) \frac{Vc_1}{L} + \\ \quad \left(\tilde{h}_q^2 - h_q^2\right) \frac{(Vc_2 - Vc_1)}{L} + \left(\tilde{h}_q^3 - h_q^3\right) \frac{(E - Vc_2)}{L} \end{cases} \quad (12)$$

Where $\tilde{R}$ is the nominal value of R without noises while R_b is the real value of R. The latter corresponds to the nominal value of R with noises.

r_1, r_2, r_3 are represented in Fig.19. As expected, r_1 and r_2 are not sensitive to this perturbation in normal conditions (R does not influence the dynamic evolution of $Vc_1$and Vc_2. While r_3 is impacted by this noise. It changes between $-0.4A/s$ and $0.4A/s$.

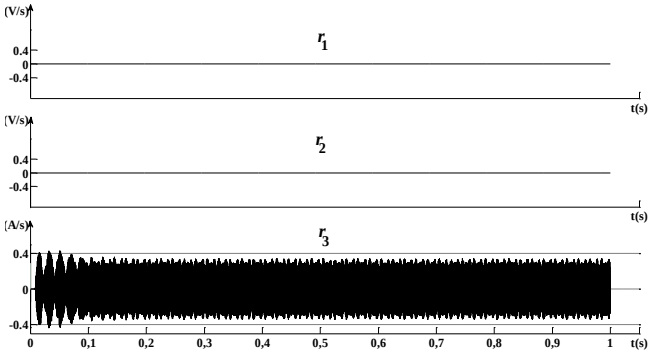


Figure 19. Set of residuals with noise corresponding to the normal conditions.

Ideally, any non-zero residual value implies a fault, which should trigger the fault isolation system. Therefore, statistical techniques are required for reliable fault detection. The fault detection system is based on a Z-test that uses the estimated variance of the residuals and a pre-specified confidence level to establish the significance of observed nonzero residuals. To cope with noise, we compute the mean and the variance at different time points (Biswas *et al*., 2003). The Z-test is a statistical inference test employed to establish the signification of the deviation. It requires the mean and standard deviation of the population, and the mean and size of the samples. These values are estimated using sliding windows over the residual for a variable. A small sliding window of size $W_1 = 5$ samples, is used to estimate the current mean $\mu_{r_i}(t)$ of the residual r_i related to the variable x_i:

$$\mu_{r_i}(t) = \frac{1}{W_1} \sum_{v=t-W_1+1}^{t} r_i(v) \quad (13)$$

We suppose the mean of the population is equal to zero, since the residual should be zero when the system is free of faults. We compute the variance from data history of the nominal residual signal over a window W_2 proceeding W_1 as an estimate of the true variance:

$$\mu'_{r_i}(t) = \frac{1}{W_2} \sum_{v=t-W_2-W_1+1}^{t-W_1} r_i(v) \quad (14)$$

$$\sigma_{r_i}(t) = \frac{1}{W_2} \sum_{v=t-W_2-W_1+1}^{t-W_1} \left(r_i(v) - \mu'_{r_i}(v)\right)^2 \quad (15)$$

The size of W_2 must contain enough of measurements in order to estimate correctly the residuals' mean and variance in the normal operating conditions and therefore to reduce the rate of false alarms. The size of W_1 must also be selected as a tradeoff between the delay of fault detection and the rate of false alarms. The size of W_2, respectively W_1, is chosen experimentally to be equal to 25, respectively 5, measurements.

Since the distribution of residuals mean is supposed to follow the normal distribution, a confidence level, α, is defined by determining the bound $\left[\mu_{r_i}^-, \mu_{r_i}^+\right]$ within which $\mu_{r_i}(t)$ is considered to correspond to normal operating conditions. $\left[\mu_{r_i}^-, \mu_{r_i}^+\right]$ is defined using Z-test table and the approximation σ_{r_i} :

$$\mu_{r_i}^- = \frac{z_{v_i}^- \sigma_{r_i}}{\sqrt{W_1}} \quad (16)$$

$$\mu_{r_i}^+ = \frac{z_{v_i}^+ \sigma_{r_i}}{\sqrt{W_1}} \quad (17)$$

For α equal to 0.95, $z_{v_i}^-$ and $z_{v_i}^+$ are equal to, respectively, -1.64 and 1.64.

The Z-test is employed in the following manner:

$\mu_{r_i}^- < \mu_{r_i} < \mu_{r_i}^+ \Longrightarrow$ No fault

Otherwise $\Longrightarrow$ Fault

Fig.20 depicts mean of residuals μ_{r_3} and the negative and positive thereshold of this residual. The mean and true variance of residual r_1 and r_2 are equal to zero. Thus its thershold is also equal to zero (μ_{r_1}, $\mu_{r_1}^-$ and $\mu_{r_1}^+$,respectively, μ_{r_2}, $\mu_{r_2}^-$and $\mu_{r_2}^+$are superposed).

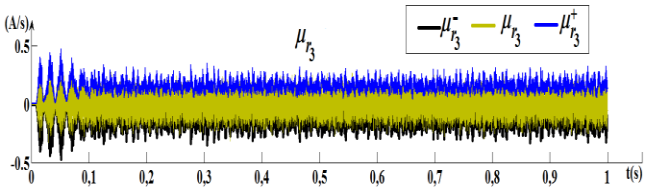


Figure 20. Set of residuals and thersholds with noise corresponding to the normal conditions.

In case of fault, Table 4 is used to achieve a local diagnosis of D_1.

Table 4. Local fault signatures generated due to the occurrence of faults in HC_1 in case of parametric noise.

SP^1	*Local signature name*	*Equivalent global fault signatures*
F_1	Sig_1^1	$\left(\frac{-I}{C_1}+\mu_{r_1}^- < \mu_{r_1} < \frac{-I}{C_1}+\mu_{r_1}^+\right)\&\left(\mu_{r_2}^- < \mu_{r_2} < \mu_{r_2}^+\right)\&\left(\frac{Vc_1}{L}+\mu_{r_3}^- < \mu_{r_3} < \frac{Vc_1}{L}+\mu_{r_3}^+\right)$
F_2	Sig_2^1	$\left(\frac{I}{C_1}+\mu_{r_1}^- < \mu_{r_1} < \frac{I}{C_1}+\mu_{r_1}^+\right)\&\left(\mu_{r_2}^- < \mu_{r_2} < \mu_{r_2}^+\right)\&\left(\frac{-Vc_1}{L}+\mu_{r_3}^- < \mu_{r_3} < \frac{-Vc_1}{L}+\mu_{r_3}^+\right)$
F_7	Sig_{71}^1	$\left(\mu_{r_1}^- > \mu_{r_1}\right)\&\left(\mu_{r_2}^- < \mu_{r_2} < \mu_{r_2}^+\right)\&\left(\mu_{r_3}^- < \mu_{r_3} < \mu_{r_3}^+\right)$
	Sig_{72}^1	$\left(\mu_{r_1}^+ < \mu_{r_1}\right)\&\left(\mu_{r_2}^- < \mu_{r_2} < \mu_{r_2}^+\right)\&\left(\mu_{r_3}^- < \mu_{r_3} < \mu_{r_3}^+\right)$
N_1	Sig_N^1	$\left(\mu_{r_1}^- < \mu_{r_1} < \mu_{r_1}^+\right)\&\left(\mu_{r_2}^- < \mu_{r_2} < \mu_{r_2}^+\right)\&\left(\mu_{r_3}^- < \mu_{r_3} < \mu_{r_3}^+\right)$

The other diagnosers D_2 and D_3 for HC_2 and HC_3 can be constructed similarly as for D_1.

4.4. Faulty conditions with parameters perturbation

In order to evaluate the proposed approach in case of noise, another scenario of fault is generated (see Fig.21). The corresponding μ_{r_1}, μ_{r_2}, μ_{r_3} for this senario are represented in Fig.22. and Fig.23. In this case, noises are observed only in r_3 at normal and faulty conditions (see zoom in Fig.24). As we said before, only r_3 is impacted by noises since the noisy parameter R is included only in dynamic evolution $\dot{I}$ of I (see (1)). To overcome this problem, a threshold is defined for each residual using Z-test. These thresholds are used during the fault detection and isolation in order to avoid the false alarms as well as the fault missed detection caused by noises.

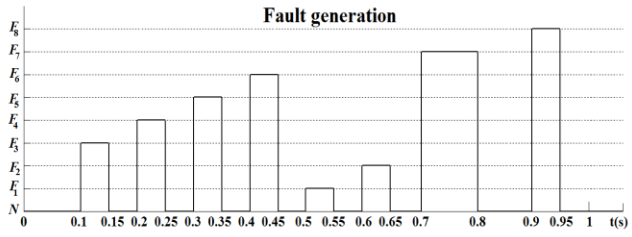


Figure 21. Time of apperance, injection, of faults during the simulation of three cell converters with noise.

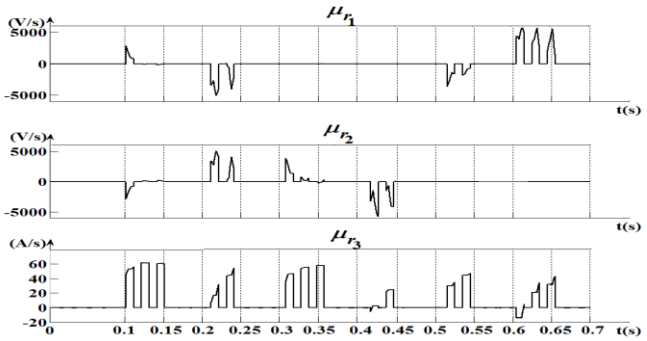


Figure 22. Residuals corresponding to generated discrete faults of Fig.21 in case of noise.

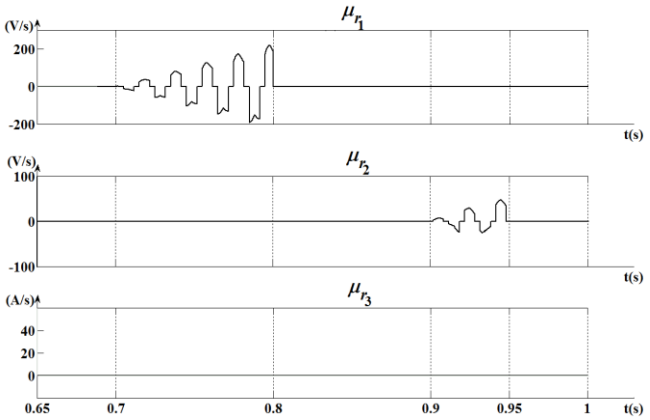


Figure 23. Residuals corresponding to generated parametric faults of Fig.21 in case of noise.

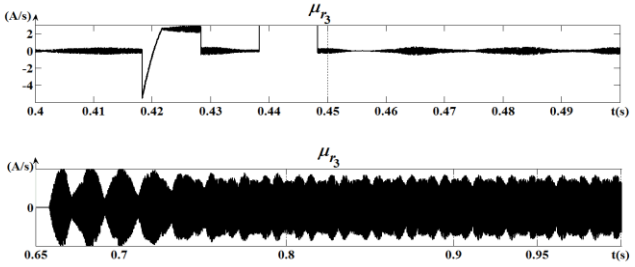


Figure 24. Zoom of residuals signals with noise corresponding to normal and faulty conditions.

Fig.24, Fig.25, Fig.26 and Fig.27 show, respectively, local decision (SP_1) of diagnoser D_1, local decision (SP_2) of diagnoser D_2, local decision (SP_3) of diagnoser D_3 and global decision (SP). The first local diagnoser D_1 is sensitive to faults of types F_1, F_2 and F_7 (diagnosis with certainty their ocurence), the second local diagnoser D_2 is sensitive to faults of types F_3, F_4, F_7 and F_8 while the third local diagnoser D_3 is sensitive to faults of types F_5, F_6 and F_8. We can conclude that the global decision indicates with certainty the occurrence of each of the generated faults regardless of the existence of noise.

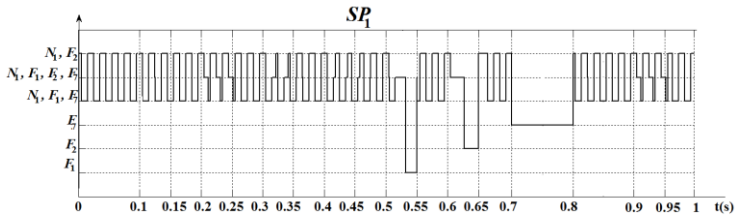


Figure 25. Local decisions (SP_1) of D_1 in case of noise.

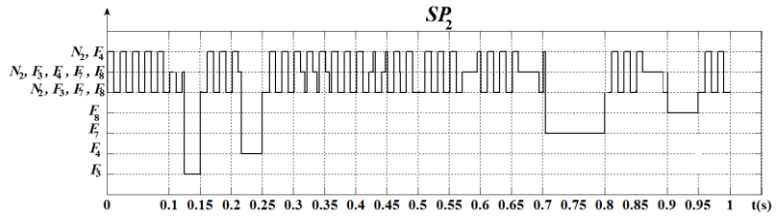


Figure 26. Local decisions (SP_2) of D_2 in case of noise.

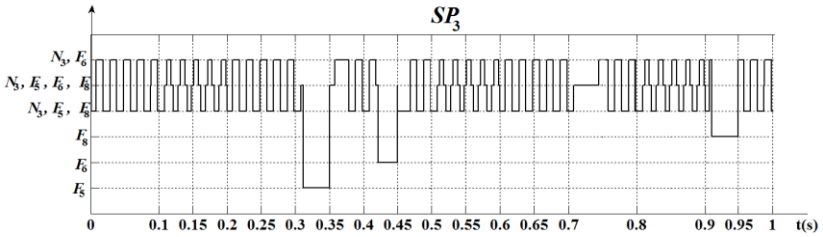


Figure 27. Local decisions (SP_3) of D_3 in case of noise.

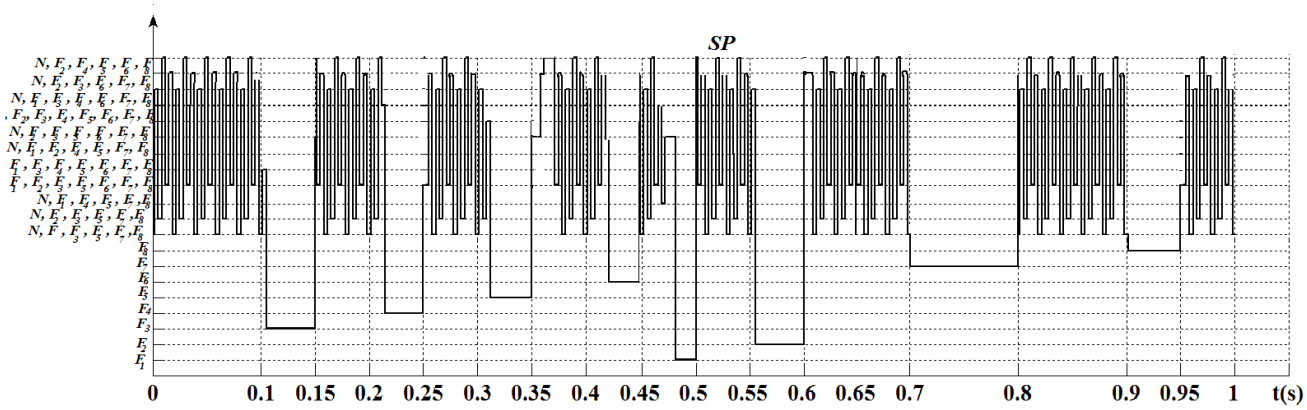


Figure 28. Global diagnosis decision issued by the coordinator in case of noise.

5. CONCLUSION

In this paper, a decentralized hybrid diagnosis approach for discretely controlled continuous systems is proposed. The elaboration of this approach is motivated by the capacity of the hybrid models to represent intrinsically the interactions between the continuous and the discrete dynamics of a system.

The originality of this work is the exploitation of the system modularity in order to reduce its complexity as well as the explosion in the number of its discrete states. To achieve that, the diagnosis task is accomplished by a set of local hybrid diagnosers. Each of the latter is responsible of the diagnosis of a specific part of the system. These local hybrid diagnosers are built without the use of the system global model but only local models. The decisions of the local hybrid diagnosers are merged using a coordinator in order to obtain a diagnosis performance equivalent to the one of a centralized diagnosis structure.

In the future work, this approach will be applied to a real

three-cell converter. Then, it will be developed to consider multiple and adjacent faults in a more general class of hybrid dynamic systems.

REFERENCES

Alavi M., Luo M., Wang D., and Zhang D., 2011. "Fault diagnosis for power electronic inverters: a model-based approach," in IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics & Drives (SDEMPED).

Arogeti S., Wang D., and Low C. B., 2010. "Mode identification of hybrid systems in the presence of fault," Industrial Electronics, IEEE Transactions on, vol. 57, no. 4, pp. 1452–1467.

Bayoudh M., Trave-Massuyés L., and Olive X.. 2006. Hybrid systems diagnosability by abstracting faulty continuous dynamics. In Proceedings of the 17th International Workshop on Principles of Diagnosis DX'06, pp. 9–15.

Bhowal P., Sarkar D., Mukhopadhyay S., and Basu A., 2007. "Fault diagnosis in discrete time hybrid systems - a case study," Information Sciences, vol. 177, pp. 1290–1308.

Biswas S., Sarkar D., Mukhopadhyay S., and Patra A., 2006. "Diagnosability analysis of real time hybrid systems", Industrial Technology. IEEE International Conference on, pp. 104–109.

Biswas G., Simon G., Mahadevan N., Narasimhan S., Ramirez J., and Karsai G., 2003. "A robust method for hybrid diagnosis of complex systems", In Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp.1125-1131.

Cocquempot V., El Mezyani T., and Staroswiecki M., 2004."Fault detection and isolation for hybrid systems using structured parity residuals," in Proceedings of the 5th IEEE/IFACASCC: Control Conference, Asian.

Daigle M., Koutsoukos X., and Biswas G., 2010. "An event-based approach to integrated parametric and discrete fault diagnosis in hybrid systems," Transactions of the Institute of Measurement and Control, vol. 32, pp. 487–510.

Derbel H., Alla H., Hadj-Alouane N. and Yeddes M., 2009. "Online diagnosis of systems with rectangular hybrid automata models," in Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing.

Defoort M., Djemai M., Floquet T., and Perruquetti W., 2011. "Robust finite time observer design for multicellular converters," International Journal of Systems Science, vol. 42, pp. 1859–1868.

Louajri H., Sayed-Mouchaweh M., and Labarre C., 2013. "Diagnoser with hybrid structure for fault diagnosis of a class of hybrid dynamic systems," Chemical Engineering Transactions, vol. 33, pp. 85–90.

Kamel T., Diduch C., Bilestkiy Y., and Chang L., 2012. "Fault diagnoses for the dc filters of power electronic converters," in Energy Conversion Congress and Exposition (ECCE), IEEE.

Rahiminejad M., Diduch C., Stevenson M., and Chang L., 2012. "Open circuit fault diagnosis in 3-phase uncontrolled rectifiers," in Power Electronics for Distributed Generation Systems (PEDG), 3rd IEEE International.

Schild A. and Lunze J., 2008. "Switching surface design for periodically operated discretely controlled continuous systems," in Hybrid Systems: Computation and Control.

Shahbazi M., Jamshidpour E., Poure P., Saadate S., and Zolghadri M., 2013. "Open- and short-circuit switch fault diagnosis for nonisolated dc-dc converters using field programmable gate array," Industrial Electronics, IEEE Transactions on, vol. 60, no. 9, pp. 4136–4146.

Trigeassou J.-C., 2011. " Diagnostic des machines électriques ". Lavoisier.

Uzunova M., Ould-Bouamama B., and Djemai M., 2012. "Hybrid bond graph diagnostic and localisation-signal signature study of three-cell converter," in Mediterranean Conference on Control & Automation (MED), Barcelona, Spain.

Zaytoon J., 2001. " Systèmes dynamiques hybrides". Hermès science publications.

BIOGRAPHIES

Hanane Louajri received her Master degree in Complex Systems Engineering section Automatic and Embedded data Processing from the University of Nancy in 2011. She received his engineering degree in Automatic and Computer Engineering from the High Engineering School "Ecole Marocaine des Sciences de l'Ingénieur" in 2011. She is currently PhD student in the High National Engineering School of Mines "Ecole Nationale Supérieure des Mines de Douai" at the Department of Automatic Control and Computer Science. Her research interests are diagnosis of hybrid dynamic systems.

Moamar Sayed-Mouchaweh received his Master degree from the University of Technology of Compiegne-France in 1999. Then, he received his PhD degree from the University of Reims-France in December 2002. He was nominated as Associated Professor in Computer Science, Control and Signal processing at the University of Reims-France in the Research center in Sciences and Technology of the Information and the Communication (CReSTIC). In December 2008, he obtained the Habilitation to Direct Researches (HDR) in Computer science, Control and Signal processing. Since September 2011, he is working as a Full Professor in the High National Engineering School of Mines "Ecole Nationale Supérieure des Mines de Douai" at the Department of Automatic Control and Computer Science (Informatique & Automatique IA).

Improved Probabilistic Remaining Useful Life Estimation in Structures: Modeling Multi-site Fatigue Cracking in Oil and Gas Service Structures

Abdallah Al Tamimi[1], Mohammad Modarres[1]

[1]University of Maryland, *College Park, Maryland, 20740, USA*

altamimi@umd.edu
modarres@umd.edu

ABSTRACT

The purpose of this research is to develop a multi-site damage probabilistic life prediction model that could be used to assess the integrity of engineering structures susceptible to fatigue in presence of neighboring cracks. Both experiments and simulation were used to produce the data required for the model development. The experiments were performed to investigate the interaction of two adjacent semi-elliptical cracks under cyclic loading. A series of tests at different loads and for different crack aspect ratios were conducted under uniaxial constant amplitude fatigue loads on API-5L grade B steel samples. Crack growth rate of two initial semi-elliptical cracks was investigated both on the sample surface and in the depth direction. Moreover, Crack growth and interaction was investigated using a simulation technique that incorporates the stress intensity factor of a single crack with an existing cracks interaction correction factor models from the literature. Finally, a Bayesian inference modeling technique is adopted to estimate the life prediction model parameters, assess any model bias and uncertainty and validate it.

1. INTRODUCTION

Oil and gas transport and storage systems are a vital cog in the oil and gas industry. Based on the nature of their functions, a combination of straight pipes, pipe-bends, dissimilar welded joints and many other parts are attached, which makes the system susceptible to many different degradation mechanisms leading to its eventual failure. This kind of system usually operates under severe conditions: internal pressure, cyclic load, internal and external environments. As a result, the combination of these different factors can lead to a potential increase in the risk of damage and unexpected fracture.

The continuously raising cost of service structures replacement, maintenance and inspection means that there are now aging systems whose continued operation requires special analysis and improved crack detection techniques. This demands continuous safety and performance improvement so that there can be increased service life of pipeline networks, maintenance, and cost control. Additionally, this necessitates early detection of a growing crack in structures like piping to prevent fracture, predict remaining useful life, schedule maintenance and reduce costly downtimes (Keshtgar & Modarres, 2013).

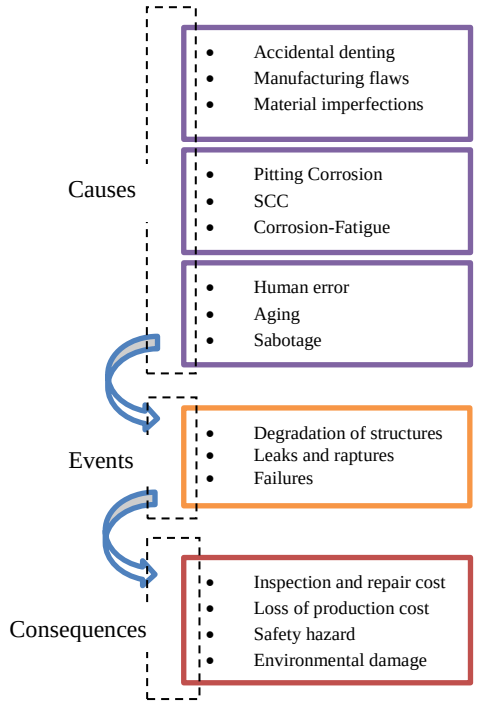


Figure 1. Causes of failures and their relative consequences

Abdallah Al Tamimi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

One of the critical failure mechanisms in engineering structures is fatigue. According to Bayley (1997), fatigue is a crack growth process that occurs under cyclic loading over the life of most engineering structures. This degradation process occurs at stresses less that the yield strength of the material until either the critical stress intensity factor is reached, leading to fracture, or until the net section yielding takes place. As crack initiation occurs in localized areas of stress concentrations, or due to environmental conditions, accumulations of pits or initial cracks are present in many structures. As these cracks interact and affect each other, the stress intensity factor ahead of the crack tip increases leading to faster crack growth rate and shorter component life. Bayley (1997) defined cracks coalescence, by several small adjacent cracks increasing in size and eventually growing together forming a single larger crack.

Numerous researchers have studied cracks interaction and coalescence including: Harrington (1995), Leek and Howard (1994, 1996), Soboyejo and Knott (1990), Kishimoto, Soboyejo, Smith, and Knott (1989), Twaddle and Hancock (1988) and O'Donoghure, Nishioka and Atluris (1984). Different assessment methods of neighboring cracks interaction and coalescence were investigated in order to identify a method that is reliable, safe and reasonably conservative and use it in order to further understand the phenomenon from a reliability/integrity stand point. Neglecting neighboring cracks interaction effect on the SIF could lead to over conservative life prediction model and assessment of structure integrity. Leek and Howard (1994) compared SIF models that does not account for cracks interactions and assume cracks re-characterization only with models that does. It was found that the safety margins achieved by re-characterization models induce overly conservative results of up to 37%.

Experimental work was performed in this research in order to investigate neighboring cracks growth rate. Different neighboring cracks geometries were investigated in order to understand the neighboring cracks dimensions effects on crack growth. Moreover, the experiments were performed under various loading conditions also in order to illuminate the role different operating conditions on the cracks growth rate. The experimental work was executed based on an improved existing technique discussed by Leek and Howard (1996) through the use of real time microscopy and digital image processing techniques of monitoring crack growth. For a more comprehensive discussion of the experimental work performed in this research, please refer to (Al Tamimi & Modarres, 2014).

Moreover, simulation efforts were also performed in order to justify the cracks interaction and coalescence behavior and explain the physics of failure aspect of the problem. The simulation provided values of the SIF at the cracks fronts and showed how it changes after each increment of growth. The simulation performed was developed by integrating Newman and Raju (1979, 1981) SIF solutions for a single semi-elliptical crack along with a cracks interaction correction factors proposed by Leek and Howard (1994).

The purpose of this study is to investigate the effect of fatigue, in presence of neighboring cracks, and integrate that into a more realistic life prediction model that could be used to predict the life of engineering structures. The need for a method of accounting for applicable and realistic cracks interaction, validated with acceptable modeling error, is the main objective of the study. This paper illustrates the modeling technique used to develop the PoF crack growth rate models. Yet, insights about the data gathering techniques and the models uncertainty quantification are also addressed mildly.

2. METHODOLOGY

The probabilistic life prediction model refers to fatigue in presence of neighboring cracks and will be developed by a procedure developed and illustrated in this paper. Two main steps are required to achieve the final modeling product. The first step is the data generation and the second step is the modeling development. In this work, the data was generated both experimentally and using simulation. Data treatment and analysis comes next in preparation for the reliability modeling. Finally, estimating the model bias and uncertainty, and validating the proposed models are considered as major steps in this model developed.

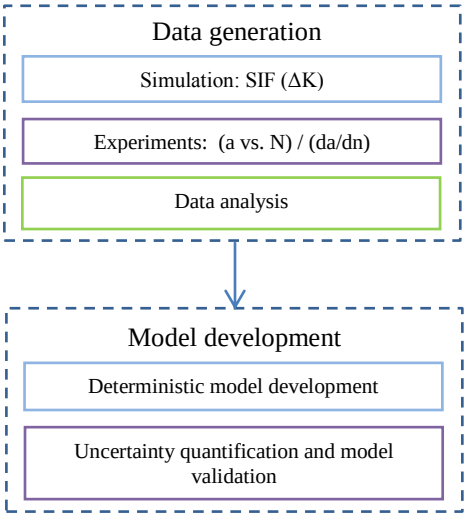


Figure 2. Modeling development steps

3. DATA GENERATION

The first step includes performing experiments in dry conditions in order to collect data about the material fatigue behavior and failure. However, the simulation focuses on understanding the SIF distribution around the cracks and

how it changes around the crack as it propagates in presence of neighboring cracks.

3.1 Experimental work

The main purpose of performing the fatigue testing was to study the fatigue properties of the material, further understand the impact of different stress levels and different crack aspect ratios on neighboring cracks, coalescence and propagation, and finally use the results for the life prediction model development.

Specimens were manufactured from an actual pipeline that was previously used in the oil and gas industry. Specimens are dog bone shaped following the ASTM E466-07, Standard Practice for Conducting Force Controlled Constant Amplitude Axial Fatigue Tests of Metallic Materials. Two initial cracks of multiple aspect ratios were machined on the sample using the electric discharge machining technique. The two cracks are semi-elliptical and co-planar simulating corrosion pits based on findings of an earlier work done by Nuhi, Abu Seer, Al Tamimi and Modarres (2011). The notches have a thickness of 0.1 mm, to assure a co-planar growth of the cracks which leads to an idealized interaction between the two cracks.

In the experimental work, the neighboring cracks were assumed to keep a semi-elliptical shape after each increment of crack growth. This assumption was made based on Nuhi et al. (2011) findings about the nature of corrosion pits shapes and geometrical development.

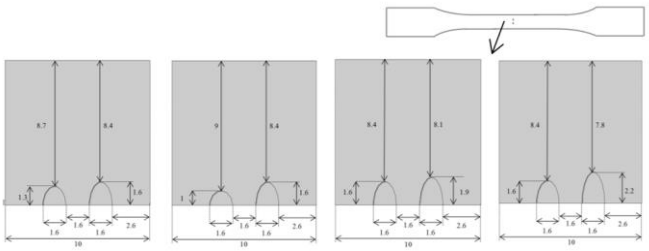


Figure 3. An illustration of the test dog bone sample and some of the notches designs used in the experimental work

Experiments were carried out at room temperature in air. An MTS fatigue-testing machine with capacity of 100 kN in tension and compression and frequency range up to 30 Hz was used. Figure 4 shows the testing setup. An optical microscope was also used to monitor the crack coalescence on the surface. The microscope is equipped with a camera to capture and save images of the specimen surface as the crack grows. Experiments are performed at constant amplitude, stress controlled cyclic loading. Frequencies of 0.2 and 2 Hz were chosen for the loading cyclic.

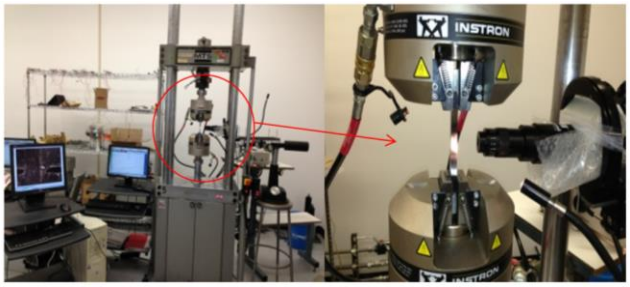


Figure 4. Experimental setup: MTS machine layout and a closer illustration of the microscope positioning

In order to gather the data required to build the probabilistic life prediction model, failed samples have to be studied and information has to be elicited. There are two main sources of information in the experimental setup used: Surface crack measurements at different number of cycles and the crack depth measurements. Linking the crack depth measurements with the recorded number of cycles at different surface crack lengths provided the scatter required for the probabilistic life prediction model.

Figure 5 shows the surface crack length and depth for one of the experiments. When enough experiments are performed and a scatter is developed, conclusions could be drawn on the applied stress and aspect ratio effect on cracks coalescence and growth.

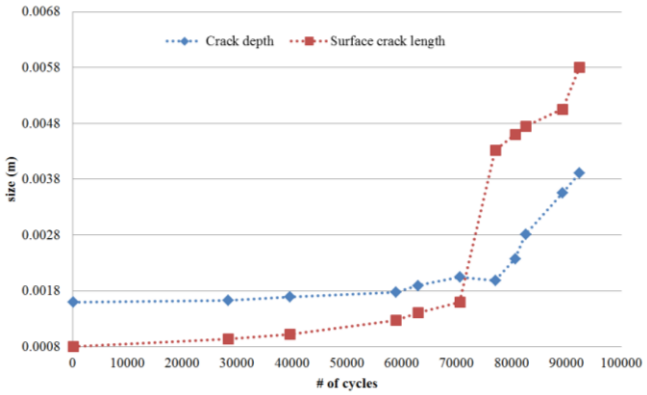


Figure 5. Example of the data elicited from the experimental work, Stress=290 MPa, Frequency=2 Hz

The experimental data scatter development is a fundamental step in the model development. An example of the data scatter developed is illustrated Figure 6 and Figure 7:

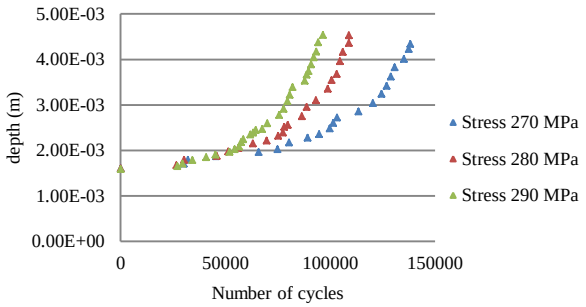


Figure 6. Effect of different stress levels on crack growth

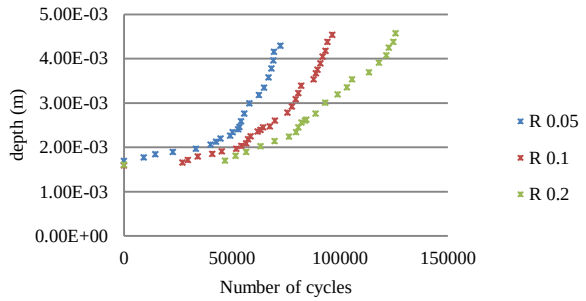


Figure 7. Effect of different loading ratios on crack growth

For more information and details about the experimental work performed in this research, please refer to (Al Tamimi & Modarres, 2014).

3.2 Simulation work

The simulation efforts were performed in order to justify the cracks interaction and coalescence and explain the physics of failure aspect. The simulation focuses on the SIF around the cracks and how it changes around the crack as it propagates in presence of neighboring cracks.

A MATLAB simulation code was developed by integrating Newman and Raju (1979, 1981) SIF solutions for a single semi-elliptical crack along with a cracks interaction correction factor empirical model by Leek and Howard (1994). The code can provide information about the SIF around a crack in presence of neighboring cracks.

The code covers a wide range of aspect ratios (a/c) and separation distance ratios (s/c). It requires certain inputs in order to find the SIF. Initial sample or plate geometry, initial cracks geometry and the development of these crack geometries are all necessary to calculate the SIF along the fatigue process.

The program can perform SIF calculation for two coplanar and identical semi-elliptical cracks geometries. However, it could be extended to cover more than two cracks. The SIF around the crack tips and front are recalculated after each increment of growth until the two cracks touch. When the cracks are predicted to touch, a single enveloping crack is immediately assumed with no further interaction factor calculations. Figure 8 illustrates both cracks front and tips.

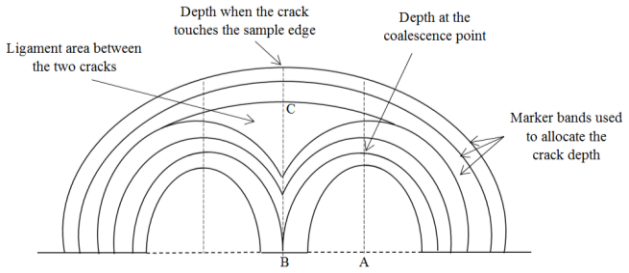


Figure 8. Cracks interaction illustration

A sample of the SIF simulation data performed for two identical cracks and its development throughout the cracks interaction and coalescence process is illustrated in Figure 9:

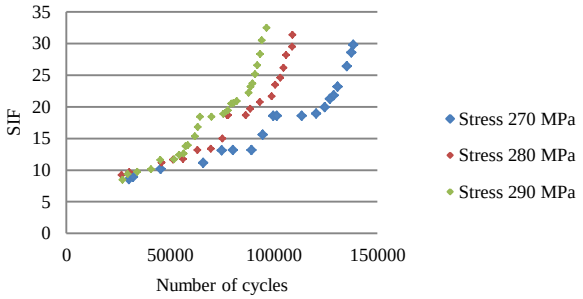


Figure 9. Crack front SIF simulation data at different stress levels

The SIF simulation data along with the experimental crack growth rate measurements will be used mainly to develop the crack growth rate models.

4. MODELING DEVELOPMENT

In this work, two models will be developed. Both models will be based on the relationship between crack growth rate and the SIF. However, different PoF base models will be used.

The first model will be constructed based on the Walker crack growth equation and the second will be based on a modified form of the Paris law equation. The two models address the same problem; however, the most suitable model with least error and uncertainties will be chosen to represent the data developed in this research. A similar modeling development strategy was used to develop both models. Table 1 summarizes and compares the two models:

Table 1. Comparison between the two crack growth models developed

Model	Walker equation model	Modified Paris law equation model
Form	$\frac{da}{dN} = \frac{C\Delta K^n}{(1-R)^{n(1-\lambda)}}$	$\frac{da}{dN} = C\Delta K^n \left(\frac{R}{R_0}\right)^m$
Variables	ΔK, LR	
Uncertain parameters	C, n, λ	C, n, m
Deterministic parameters	/	R_0
Data sources	Experimental data (da/dn values)	
	Simulation data (ΔK values)	

The main sources of data scatter in this work are fatigue experiments and simulation. Producing usable results that can appropriately capture not only the effect of time, applied stress levels, but also the effect of cracks aspect ratios is one of the main objectives of this work. The models main variables are the SIF and loading ratio, however, other variables like the applied stress are still considered in this work. Although the stress term is not apparent in the PoF model, yet it is embedded in the SIF term. The same applies for other variables like the neighboring cracks dimensions. Experimental data has been split into two different sets:

1. Deterministic model development data set
2. Uncertainty quantification and model validation data set

The use of each set, which are independent from each other, is represented in Figure 10. Each model development stage requires an independent data set which will minimize the bias in the model development.

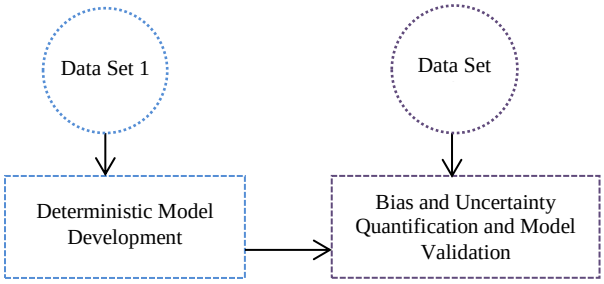


Figure 10. Model development stages

4.1 Deterministic model development

As deliberated earlier, the modeling efforts discussed developing two PoF crack growth models. The first proposed model is based on the Walker equation having the illustrated mathematical representation in Equation 1:

$$\frac{da}{dn} = f(\Delta K, R|_{C,\,n,\,\lambda}) \quad (1)$$

On the other hand, the second PoF crack growth rate model is slightly different as it is based on the Paris law equation. However, a correction factor term was added to the equation to account for the effect of loading ratio on the crack growth rate. The mathematical form of this model is illustrated in Equation (2):

$$\frac{da}{dN} = f(\Delta K, R|_{C,\,n,\,m,\,R_0}) \quad (2)$$

The same deterministic model development methodology is followed when developing both forms of the crack growth rate model. However, and for illustration purposes, the procedure will be explained and illustrated based on the Walker equation PoF crack growth model.

In order to shape the final form of the deterministic model, a proper evaluation of the model uncertain parameters is required. The proposed model parameters C, n and λ have been estimated from generic data available in literature, experiments and simulations developed in this research.

As there are an infinite number of possible fatigue experiments and simulations to perform to fully understand the nature of interactions between neighboring cracks, Obtaining data for such failure mechanism has proven to be difficult, time consuming and very expensive. Yet, a great analytical tool that enables the integration of new evidence with the existing prior knowledge and produces an updated knowledge of the uncertain model parameters is Bayes' theorem. As such, the Bayesian estimation method was applied in this research to estimate the uncertain parameters C, n and λ.

A Bayesian inference will be used to develop the deterministic model as it is a powerful mathematical tool that could estimate/update the model parameters with minimum amount of data. The Bayesian inference is a method used to update a given state of knowledge based on new given evidence. A summary of this process is illustrated in Figure 11:

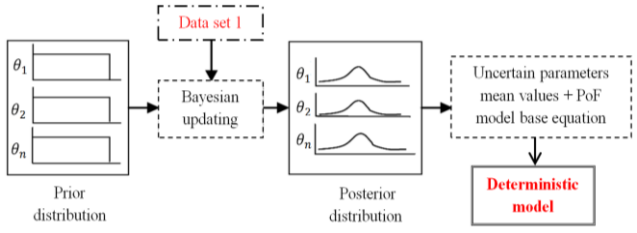


Figure 11. Deterministic model development (Azarkhail & Modarres, 2012)

In the Bayesian inference, a subjective prior probability distribution (pdf) of each of the model uncertain parameters $f_o(C, n, \lambda)$ was defined based on a comprehensive literature search. For example, different researchers like Neves

Beltrao, Castrodeza, & Bastian (2010), Shi, Chen and Zhang (1999), Fernandes (2002) and Hamam, Pommier and Bumbieler (2007) have investigated crack growth in carbon steel materials and provided quantifications of the Paris law equation coefficients. Such quantifications was be used as priors in the Bayesian inference performed to update the knowledge of the model uncertain parameters. When there was no prior information available in the literature about a certain uncertain parameter, a non-informative uniform distribution was assumed.

Subsequently, this prior was combined with the evidence data in the form of a likelihood function. The likelihood equation of the crack growth rate was assumed to follow a normal distribution and is illustrated in Equation 3:

$$L(\mathrm{C},\mathrm{n},\lambda,\sigma|Data)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{\frac{da}{dN_i}-\left(\frac{C\Delta K_i{}^n}{(1-R_i)^{n(1-\lambda)}}\right)}{2\sigma^2}} \quad (3)$$

The result is an updated state of knowledge identified as the posterior distribution, f(C, n, λ, σ |Data). This process is shown mathematically in Equation (4):

$$f(\mathrm{C},\mathrm{n},\lambda,\sigma|Data)=\frac{L(\mathrm{C},\mathrm{n},\lambda|Data)f_0(\mathrm{C},\mathrm{n},\lambda)}{\int_\theta L(\mathrm{C},\mathrm{n},\lambda|Data)f_0(\mathrm{C},\mathrm{n},\lambda)} \quad (4)$$

To accomplish this task, WinBUGS software program was employed to run the Bayesian analysis. In line with Spiegelhalter, Thomas, Best and Lunn (2003) the WinBUGS program is a windows-based environment for MCMC simulation. A wide variety of modeling applications could benefit from using such software. This program has been previously reported to be used in uncertainty management according to Azarhkail and Modarres (2007) as well as accelerated life testing data analysis and has proved to be a reliable tool for such calculations. In this research the WinBUGS platform was used for Bayesian updating and related numerical simulations. After running the developed WinBUGS code, a posterior knowledge of the uncertain parameters C, n and λ is obtained.

4.2 Uncertainty quantification and model validation

As the proposed model uncertain parameters C, n and λ were initially estimated using the information available in the literature. However, these estimations require further validation before it can be deployed for additional analysis. Hence, Bayesian approach was utilized to investigate the validity of this prior estimation and then was applied to the updating procedure.

In this step, a more comprehensive model bias and uncertainty analysis is performed. A method developed by Azarkhail and Modarres (2007) and Ontiveros, Cartillier and Modarres (2010) and modified and used later by Keshtgar (2014) to quantify the model uncertainties will be used. However, a different set of evidence data is used for this purpose. The bias and uncertainty quantification is based on comparing the model predictions with the experimental results as illustrated in Figure 12:

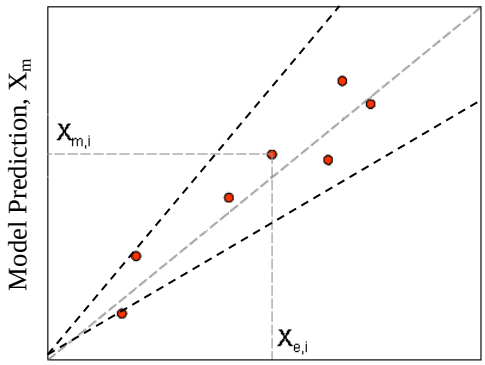


Figure 12. Deterministic model predictions compared to experimental results (Azarkhail, Ontiveros, & Modarres, 2009)

If the model predictions perfectly matched the experimental results, then all the points would lie exactly on the dotted line which is not highly probable. This is because of the uncertainties and possible bias in both the model predictions and the experimental measurements.

In this research, the model prediction and experimental result are considered to be estimations of the crack growth rate (da/dN), given some error as shown in Equations 5 and (6):

$$\frac{da/dN_i}{da/dN_{e,i}}=F_{e,i}\ ;\ F_e\sim LN(b_e,s_e) \quad (5)$$

$$\frac{da/dN_i}{da/dN_{m,i}}=F_{m,i}\ ;\ F_m\sim LN(b_m,s_m) \quad (6)$$

As the modeling addresses crack growth values, then the model outcome is always expected to be a positive value, for that reason, a multiplicative error model is assumed. Moreover, the error is assumed to be distributed log-normally for the same reason.

As the true value of the crack growth rate da/dN_i is unknown, Equations 5 and 6 are combined yielding the following equations:

$$F_{e,i}\left(da/dN_{e,i}\right)=F_{m,i}\left(da/dN_{m,i}\right) \quad (7)$$

$$\frac{da/dN_{e,i}}{da/dN_{m,i}}=\frac{F_{m,i}}{F_{e,i}}=F_{t,i} \quad (8)$$

Assuming independency of F_m , F_e then:

$$F_t\sim LN\left(b_m-b_e,\sqrt{{s_m}^2+{s_e}^2}\right) \quad (9)$$

The likelihood used in the Bayesian inference is illustrated in Equation 10:

$$L\left(F_{t,i}, b_e, s_e \middle| b_m, s_m\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\left(F_{t,i}\right)\sqrt{{s_m}^2 + {s_m}^2}} e^{-\frac{\left[ln\left(F_{t,i}\right)-\left(b_m-b_e\right)\right]^2}{2\left({s_m}^2+{s_m}^2\right)}} \quad (10)$$

Finally, the Bayesian inference is performed, where Equation 11 shows the relation between the posterior distribution of the model parameter with the likelihood function and the prior evidence.

$$f\left(b_m, s_m \middle| F_{t,i}, b_e, s_e\right) = \frac{L\left(F_{t,i}, b_e, s_e \middle| b_m, s_m\right) f_0(b_m, s_m)}{\int\left(L\left(F_{t,i}, b_e, s_e \middle| b_m, s_m\right) f_0(b_m, s_m)\right)} \quad (11)$$

The data used in this step of the analysis must be data independent of the data used in the model development step.

Quantifying the bias and uncertainty is considered also a validation of the models proposed. Assuming the model-based predicted crack growth rate is da/dn_m, the true crack growth rate prediction can be estimated by multiplying da/dn_m by the estimated F_m:

$$\frac{da}{dN}_{true} = \frac{da}{dN}_m . F_m \quad (9)$$

The model prediction results will be modified using the resulted bias distribution which can be estimated by a lognormal distribution:

$$\frac{da}{dN}_{true} \sim LN\left(ln\left(\frac{da}{dN}_m\right) + b_m , s_m\right) \quad (10)$$

5. CONCLUSION

Many different degradation mechanisms act on engineering structures causing all different types of flaws and imperfections which eventually cause failure affecting the integrity of many critical systems. Given that capturing all degradation mechanisms would be a challenging task, this work focuses on fatigue as the main failure mechanism. Fatigue is one of the degradation failure mechanisms that accelerate the failure of engineering structures. However, other critical failure mechanisms like corrosion, stress corrosion cracking and creep are also of great importance and should not be disregarded. Moreover, factors like the type of material and the loading conditions plays a crucial role in the degradation rate of the structure. So in order to have a best estimate of the structure reliability, these factors should be taken into consideration.

This paper provides a summary of the methodology used to develop a PoF life prediction model that addresses fatigue of neighboring cracks. This summary includes highlights of the data gathering techniques. Moreover, it discusses the possible forms of the life prediction model and how to identify its uncertain parameters using Bayesian inference.

One of the main outcomes of this research is probabilistic life prediction models that address fatigue as a failure mechanism in presence of neighboring cracks. This kind of models could be used in assessing the integrity of certain engineering structure and serve as a guide for maintenance planning. This kind of models could be continuously updated along the spectrum of the structure life by adding more evidence gathered from monitoring its health and operation.

Both experiments and simulation were used to produce the data required for the model development. The experiments were performed to investigate the interaction of two adjacent semi-elliptical cracks of variable dimensions under different cyclic loading conditions. This will allow the model to capture a wide range of applications and make it more realistic. A series of tests at different loads and loading ratios were conducted under uniaxial constant amplitude fatigue loads on API-5L grade B steel samples. Crack growth rate of two initial semi-elliptical cracks was investigated both on the sample surface and in the depth direction.

Furthermore, the simulation was performed to understand the SIF behavior around a crack when it is surrounded by neighboring cracks providing a better understanding of the failure mechanism and justifying its behavior under different loading conditions. Crack growth and interaction was investigated using a simulation technique that incorporates the stress intensity factor of a single crack with an existing cracks interaction correction factor models from the literature.

The Bayesian approach was used to construct the life prediction models using both the experimental and simulation data and estimate their parameters. Uncertainties about the structure of the model and its parameters were also characterized in this work.

ACKNOWLEDGEMENT

The authors would like to thank Abu Dhabi National Oil Company (ADNOC), Abu Dhabi Company for Onshore Oil Operations (ADCO) and the Petroleum Institute (PI).

NOMENCLATURE

C	Paris law empirical constant
n	Paris law empirical constant
LR	Loading ratio
λ	Empirical constant that indicates the influence of the loading ratio on the fatigue crack growth in different materials
m	Uncertain parameter in the Paris law loading ratio correction factor
R_0	Deterministic parameter in the Paris law loading ratio correction factor
N	Number of cycles
θ_i	Uncertain parameter
da/dn_i	Crack growth rate true value
$da/dn_{e,i}$	Crack growth rate value obtained experimentally
$da/dn_{m,i}$	Crack growth rate value obtained from the model developed
$da/dn_{true,I}$	Corrected crack growth rate value
F_e	The multiplicative error of the experimental crack growth value with respect to the true value
F_m	The multiplicative error of the model crack growth prediction with respect to the true value
b_e	The experimental mean multiplicative error
s_e	The Standard deviation of the experimental multiplicative error
b_m	The model mean multiplicative error
s_m	The standard deviation of the model multiplicative error
F_t	The multiplicative error of experiment with respect to model prediction

REFERENCES

Al Tamimi, A., & Modarres, M. (2014). Coalescence and Growth of Two Semi-elliptical Coplanar Cracks in API-5L Grade B steel . *SEM 2014 Annual Conference & Exposition on Experimental and Applied Mechanics.* Greenville, SC, USA.

Azarhkail, M., & Modarres, M. (2007). Markov chain Monte Carlo simulation for estimating accelerated life model parameters. *Annual Reliability and Mainitainability Symposium (RAMS) Proceedings.* Orlando.

Azarkhail, M., & Modarres, M. (2007). A Novel Bayesian Framework for Uncertainity Management in Physics-Based Reliability Models. *ASME International Mechanical Engineering Congress and Expositon* . Seattle, Washignton, USA.

Azarkhail, M., & Modarres, M. (2012). The Evolution and History of Reliability Engineering: Rise of Mechanistic Reliablity Modeling. *International Journal of Performability Engineering , 8*, 35-47.

Azarkhail, M., Ontiveros, V., & Modarres, M. (2009). A Bayesian Framework for Model Uncertainity Consideration in Fire Simulation Codes. *17th International Conference on Nuclear Engineering* . Brussels.

Bayley, C. (1997). *Parametric Investigation on the Coalescence of Coplanar Fatigue Cracks, Masters' thesis.* Ottawa: Carelton University .

Fernandes, J. (2002). *Uma metodologia para a analise e modelagem, PhD thesis.* Rio de Jeneiro, Brasil: PUC.

Hamam, R., Pommier, S., & Bumbieler, F. (2007). Variable amplitude fatigue crack growth, experimental results and modeling . *29*(1634-1646).

Harrington, D. (1995). *Fatigue Crack Coalescence and Shape Development and Experimental Invistigation, Masters Thesis.* Carleton University, Department of Mechanical and Aerospace Engineering .

Keshtgar, A. (2013). *Acoustic emission-based structural health management and prognostics subject to small fatigue cracks, PhD thesis* . College Park: University of Maryland.

Keshtgar, A., & Modarres, M. (2013). Probabilistic model developement for fatigue crack detection using acoustic emission technology. *International Topical Meeting on Probabilistic Safety Assesement and Analysis.* Columbia, SC: ANS PSA.

Kishimoto, K., Soboyejo, W., Smith, R., & Knott, J. (1989). A Numerical Invistigation of the Interaction and Coalescence of Twin Coplanar Semi Elliptical Fatigue Cracks. *International Journal of Fatigue , 11*(2), 91-96.

Leek, T. (1990). *The Interaction and Growth of Two Surface Cracks Under Fatigue Loading, PhD Thesis.* UK: University of Sheffield.

Leek, T., & Howard, I. (1994). Estimating the Elastic Interaction Factors of Two Coplanar Surface Cracks Under Mode I load. *International Journal of Pressure Vessels and Pipeing , 60*, 307-321.

Leek, T., & Howard, I. (1994). Rules for the Assessment of Interacting Suface Cracks Under Mode I Load. *Internationl Journal of Pressure Vessels and Piping , 60*, 323-339.

Leek, T., & Howard, I. (1996). An Examination of Methods of Assessing Interacting Surface Cracks by Comparison with Experimental Data. *International Journal of Pressure Vessels and Piping, 68*, 181-201.

Neves Beltrao, M., Castrodeza, E., & Bastian, F. (2010). Fatigue crack propagation in API 5L X-70 pipeline steel longitudinal welded joints under constant and variable amplitude. *Fatigue and Fracture of Engineering Materials and Structures, 34*, 321-328.

Newman, J., & Raju, I. (1979). Stress intensity Factors for a Wide Range of Semi-Elliptical Suface Cracks in

Finite Width Plates. *Engineering Fracture Mechanics, 11*.

Newman, J., & Raju, I. (1981). An Emperical Stress Intensity Factor Equation for the Surface Crack. *Engineering Fracture Mechanics, 15*.

Nuhi, M., Abu Seer, T., Al Tamimi, A., & Modarres, M. (2011). Reliability anlysis for degradation effects of pitting corrosion in carbon steel pipes. *International Conference on Mechanical Behaviour of Materials . 10.* Como: Elsevier.

O'Donoghure, T., Nishioka, P., & Atluris, N. (1984). Multiple Surface Cracks in Pressure Vessels. *Engineering Fracture Mechanics, 20*(3), 545-560.

Ontiveros, V., Cartillier, A., & Modarres, M. (2010). An Integrated Methodology for Assessing Fire Simulation Code Uncertainity. *Nuclear Science Engineering , 166*, 179-201.

Shi, Y., Chen, B., & Zhang, J. (1999). Effects of welding residual stresses on fatigue crack growth behaviour in butt welds of a pipline steel. *Engineering Fracture Mechanics, 36*, 893-902.

Soboyejo, W., & Knott, J. (1990). Fatigue Crack Propagation of Copalanr Semi-elliptical Cracks in Pure Bending. *Engineering Fracture Mechanics, 37*(2), 323-340.

Spiegelhalter, D., Thomas, A., Best, N., & Lunn, D. (2003). *WinBUGS 1.4 manual* .

Twaddle, J., & Hancock, B. (1988). The Development of Cracks by Defect Coalescence. *Engineering Materials Advisory Service*, 185-198.

BIOGRAPHIES

Abdallah Al Tamimi is a mechanical engineer and a current PhD student in the University of Maryland, College Park, USA. He earned his mechanical engineering degree from the Petroleum Institute, Abu Dhabi, United Arab Emirates. Also, he earned his Master's in Reliability engineering from the University of Maryland, College Park, United States. Over the last four years, he has worked in a variety of professional organizations in both academia and industry. He worked in Abu Dhabi Marine Operating Company (ADMA-OPCO) as a mechanical supervisor in the oil and gas industry. Afterwards, he made his transition to academia by working as a teaching assistant at the Petroleum Institute. He started his PhD studies in reliability engineering in 2011 after he was granted a fellowship by Abu Dhabi National Oil Company (ADNOC).

Mohammad Modarres is the Director of Reliability Engineering and Minta Martin Professor of Engineering in the University of Maryland, College Park, United States. He specializes in probabilistic risk assessment and management, uncertainty analysis, probabilistic physics of failure and probabilistic fracture mechanics modeling. He earned his B.S. degree from Tehran Polytechnic in mechanical engineering, a Master's degree from Massachusetts Institute of Technology in mechanical engineering and a PhD from Massachusetts Institute of Technology in nuclear engineering. He has served as a consultant or board member to several governmental agencies including the Nuclear Regulatory Commission, Department of Energy, National Academy of Sciences, and several national laboratories in areas related to nuclear safety, probabilistic risk assessment, probabilistic fracture mechanics and physics of failure. He has over 300 papers in archival journals and proceedings of conferences. He has published a number of textbooks, edited books and book chapters in various areas of nuclear safety, risk and reliability engineering. He is a University of Maryland Distinguished Scholar-Teacher and a fellow of the American Nuclear Society.

Imaging and Information Processing of Pitting-Corroded Aluminum Alloy Panels with Surface Metrology Methods

Honglei Li[1], Margaret R. Garvan[2], Jiaming Li[3], Javier Echauz[4], Douglas Brown[5], George J. Vachtsevanos[6]

[1, 2, 3, 6] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA*

honglei.li@gatech.edu
mgarvan3@gatech.edu
jli339@gatech.edu
gjv@ece.gatech.edu

[4] *JE Research, Inc., 170 Wentworth Terrace, Alpharetta, GA 30022, USA*

echauz@ieee.org

[5] *Analatom, Inc.,562 E. Weddell Dr. Suite 4, Sunnyvale, CA 94089-2108, USA*

Doug.Brown@analatom.com

ABSTRACT

It has been established that corrosion is one of the most important factors causing structural deterioration, loss of metal, and ultimately decrease of product performance and reliability. Corrosion monitoring, accurate detection and interpretation are recognized as key enabling technologies to reduce the impact of corrosion on the integrity of critical aircraft and industrial assets. Interest in corrosion measurement covers a broad spectrum of technical approaches including acoustic, electrical and chemical methods. Surface metrology is an alternative approach used to measure corrosive rate and material loss by obtaining surface topography measurement at micrometer levels. This paper reports results from an experimental investigation of pitting corrosion detection and interpretation on aluminum alloy panels using 3D surface metrology methods, image processing and data mining techniques. Sample panels of AA 7075-T6, an aluminum alloy commonly used in aircraft structures, were coated on one side with a corrosion-protection coating and assembled in a lap-joint configuration. Then, a series of accelerated corrosion testing of the lap-joint panels were performed in a cyclic corrosion chamber running ASTM G85-A5 salt fog test. Panel surface characterization was evaluated with laser microscopy and stylus-based profilometry to obtain global and local surface images/characterization. Promising imaging and surface features were extracted and compared between the uncoated and coated panel sides, as well as on the uncoated sides under different corrosion exposure times. In the evaluation process, image processing, information processing and other data mining techniques were utilized. Information processing involves the steps of feature or Condition Indicator extraction and selection. The latter step addresses the problem of selecting those features that are maximally correlated with the actual corrosion state, for the purpose of corrosion detection, localization, quantification and state estimation. The results, verified by mass loss data, confirmed the contention that pits at the panel surfaces formed as a result of electrochemical corrosion attack, and showed that deteriorating pitting corrosion attack correlates with increasing corrosion exposure times. This study is a first step in the process of understanding, assessing and responding to the pitting corrosion and ultimately preventing material failure to insure aircraft structural integrity.

Honglei Li et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Every year, corrosion is responsible for billions of dollars loss in structural deterioration, loss of metal, and ultimately decreased product performance and reliability. Pitting corrosion is one of the most prevalent forms of localized corrosion, a dangerous phenomenon because of its rapid damage growth rate, and the difficulty to detect it and predict its evolution. The pitting attack is highly localized and is usually in the form of holes that can penetrate inwards extremely rapidly and ultimately damage the structure by either perforating the material or developing into cracking due to stress corrosion (Rao & Rao, 2004). It is thus essential to insure the critical assets' integrity and operational safety by condition-based monitoring, early

detection, interpretation and prediction of pitting attack. Many research efforts have been reported in the past addressing this critical issue (Frankel, 1998; Szklarska-Smialowska, 1999; Huang & Frankel, 2006; Pereira, Silva, Acciari, Codaro & Hein, 2012). However, undeniably, well-recognized global corrosion measurements, such as weight loss and wall thickness reduction, cannot offer an appropriate and trustworthy way to interpret the pitting corrosion due to its localized attack nature. To address the need for accurate detection, interpretation and prediction of pitting corrosion, this paper proposes the use of surface metrology methods together with image and information processing techniques that take advantage of accurate and thorough testing evidence.

1.1. Motivation

Detection, localization and quantification of corrosion in complex structures over large, partially accessible areas are of growing interest in the aerospace industries. Traditionally, conventional ultrasonics and eddy current techniques have been used to precisely measure the thickness reduction in aircraft structures. However, the scanning may become impossible when the area of inspection is inaccessible. Upon this need, there has been a number of undergoing research using guided wave tomography technique to screen large areas of complex structure for corrosion detection, localization (Clarke, 2009) and defect depth mapping (Belanger, Cawley & Simonetti, 2010). However, due to the nature of ultrasonic guided wave, this technique is vulnerable to environmental changes, especially to temperature variation and surface wetness occurrence (Li, Michaels, Lee, & Michaels, 2012), and the precision of corrosion defect depth reconstruction is restricted by sensor network layout, structure complexity, and other factors, which limits the scope of the field application.

On the other hand, in the field of surface metrology, there are various techniques for quantitative characterization of surface topology, generally categorized into contact and non-contact measuring methods, which are promising techniques for corrosion, especially localized corrosion detection and characterization. The traditional contact profilometry has the merits of reliable measurement and low cost, and the disadvantage of low speed, and resolution and applicable surface limitation. On the contrary, the optical non-destructive metrology has the merits of high speed, high profiling resolution and non-destructiveness, and the disadvantage of high scatter noise and high cost.

1.2. Methodology

In this paper, we take advantage of both contact and non-contact surface metrology techniques to obtain 2D and 3D images/profiles for accurate characterization of pitting corrosion attack in AA7075-T6 aluminum alloy panels; extract and select promising morphologic and texture features from images, as well as profile features from surface measurements. Note that both global and local metrology measurements and image/profile data analysis approaches are adopted here for the purpose of accurate detection, localization and interpretation of pitting corrosion. To facilitate early detection of corrosion attack, initial testing procedures, data acquisition and feature extraction focus on global approaches, i.e., the whole panel area is viewed as the target for data collection and analysis. After the corrosion detection, localized studies are adopted where imaging studies, for example, focus on small areas of the global image where corrosion initiation is suspected, localized, or prone to spread more rapidly than other areas. The highlight of this work is the utilization of 3D surface metrology testing tools and novel image/information processing methods to study the features of interest for corrosion analysis.

The remainder of the paper is organized as follows. Section 2 introduces the procedures of accelerated corrosion testing. Section 3 describes the facilities and procedures of 3D surface metrology testing for imaging/ characterization data acquisition. Section 4 introduces the methodologies used in corrosion data mining, including image pre-processing, feature extraction and feature selection. Section 5 presents the analysis results for pitting corrosion detection, localization and interpretation. Section 6 concludes the paper with a summary of future work.

2. ACCELERATED CORROSION TESTING

2.1. Testing Preparation

New aluminum alloy AA7075-T6 and AA2024-T3 samples were cut to dimension of 6’×3’×1’ and uniquely marked with stencil stamps close to the edge of both faces of the sample. A sample panel is shown in Figure 1. The samples were then cleaned using an alkaline cleaner, TURCO 4215 NC-LT – 50 g/L for 35 min at 65℃. Afterwards, the samples were rinsed with Type IV reagent grade deionized water and immersed in a solution of 20% (v/v) nitric acid for 15 minutes. The samples were then rinsed again in the deionized water and air dried. The weights were recorded to the nearest fifth significant figure and the samples were stored in a desiccator. After massing, the samples were assembled in a lap-joint configuration as shown in Figure 2, and coated with 2 mils of epoxy-based primer and 2 mils of polyurethane.

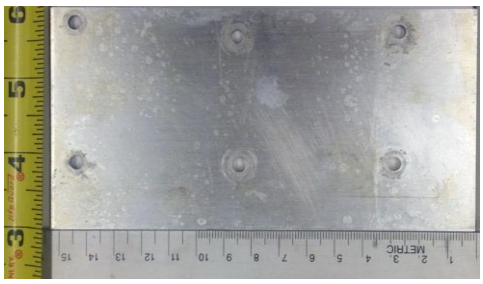


Figure 1. Corrosion panel sample on the uncoated side with 6 through rivet holes, AA 7075-T6.

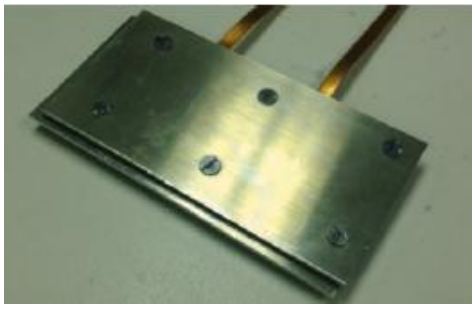


Figure 2. AA7075-T6 and AA2024-T3 lap joint assembly.

2.2. Cyclic Corrosion Testing

Corrosion tests were performed in a cyclic corrosion chamber running a modified B117 salt-fog test, specifically, the ASTM G85-A5 test. This test consisted of two one hour steps. The first step involved exposing the samples to a salt fog for a period of one hour at 25 ℃. The electrolyte solution composing the fog was 0.05% sodium chloride and 0.35% ammonium sulfate in deionized water. This step was followed by a dry-off step, where the fog was purged from the chamber while the internal environment was heated to 35 ℃. Electrical connections for the flex sensors were made to an AN110 positioned outside the sealed chamber by passing extension cables through the bulkhead in the chamber. Temperature and relative humidity were acquired at 1-minute intervals.

At the conclusion of this experiment, lap joints were removed from the environmental chamber and disassembled. Following disassembly, the polyurethane and epoxy coatings on the aluminum panels were removed by placing them in a solution containing methyl ethyl ketone. After a 30-minute immersion the panels were removed and rinsed with deionized water. These panels were again alkaline cleaned with a 35-minute immersion into a constantly stirred solution of 50 g/l Turco 4215 NC-LT at 65 ℃. This was followed by a deionized water rinse and immersion into a 90 ℃ solution of 85% phosphoric acid containing 400 g/l chromium trioxide for 10 minutes. Following phosphoric acid treatment, the panels were rinsed with deionized water and placed into a 20% nitric acid solution for 5 minutes at 25 ℃. Plates were then rinsed with deionized water, dipped in ethanol, and dried with a heat gun. This cleaning process was repeated until mass values for the panels stabilized. These values were then compared with values predicted from the results from surface metrology image processing.

This experiment ran over a period of 286 hours, where the environment inside the chamber was varied in temperature and humidity to promote corrosion. Panels 1-3 were removed 133, 209 and 286 hours from the experiment, respectively, preparing for the surface metrology testing. Detailed explanation of the accelerated corrosion testing is introduced in a complementary paper.

3. 3D SURFACE METROLOGY FOR CORROSION ANALYSIS

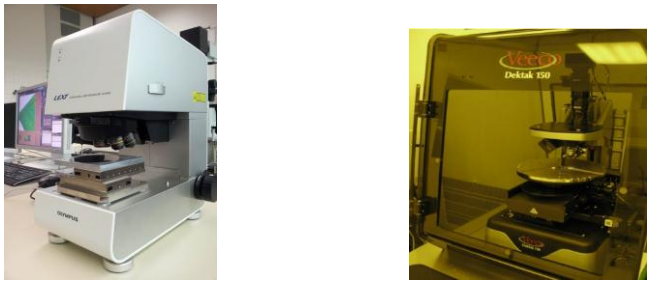
Surface metrology is the measurement of small-scale features on surfaces, which can be realized through contact or non-contact instruments as introduced before. Here, we utilize state-of-the-art laser microscopy and stylus-based profilometry surface measurement equipment to obtain 2D and 3D images and characterization data of corroded surfaces and extract from them relevant information that assists in corrosion detection and interpretation.

In this preliminary work, for the illustration of methodology, our study focuses on the corrosion behavior of AA 7075-T6 panels of 3 different corrosion exposure times. AA2024-T3 panels from the corresponding lap joints will be examined in the future work. In this testing, we use a confocal laser microscope and a stylus-based profilometer together to achieve a thorough examination of the corroded panels with rivet holes. The Olympus LEXT OLS4000 3D Laser Confocal Microscope, as shown in Figure 3(a), is designed for nanometer level imaging, 3D surface characterization and roughness measurement. Magnification ranges from 108x to 17,280x. The Bruker's Dektak 150 Stylus Profilometer, as shown in Figure 3(b), is a traditional 2D tactile profilometer. With the programmable map scan capability and the post-processing software, it allows for large area 3D topography coverage. The combination of the two surface metrology tools facilitates both localized and global characterization of a corroded panel at various resolution scales.

The surface metrology testing scheme is summarized as below:

1) Global characterization:

   - The laser microscope can provide large area 2D microscopy imaging by stitching adjacent images.
   - The stylus profilometer can provide large flat area (i.e., surface without rivet holes) 3D map scan imaging. A schematic of the area the profilometer covers in a 3D map scan for a typical panel is shown in Figure 4.



(a) (b)

Figure 3. Surface metrology measuring tools: (a) Olympus LEXT OLS4000 3D Laser Confocal Microscope, (b) Bruker Dektak 150 Stylus Profilometer.

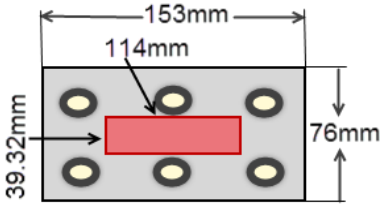


Figure 4. The area (in red) the profilometer covers in a 3D map scan.

2) Local characterization:

- After the corrosion detection and localization, the laser microscope can provide a close look at the 3D topography of the analyzed surface areas.

First, for corrosion detection and quantification, global characterization was performed through both the microscope and the profilometer for each panel corresponding to a specific corrosion exposure time: while the microscope provided whole-panel 2D imaging, the stylus profilometer provided contact 3D map scan of the general central region without rivet holes. Next, local 3D characterization of areas of interest was conducted through the microscope. The further surface analysis was performed based on the local 3D characterization and a list of surface parameters was calculated for further processing.

4. CORROSION DATA MINING

An important and essential component of the corrosion detection and interpretation architecture involves image/characterization data pre-processing and data mining aimed to extract and select useful and relevant information from raw data. In the proposed architecture, the most important components supporting the implementation of the framework are feature extraction and selection. Features are the foundation for the fault/corrosion detection and interpretation scheme. Feature extraction and selection processes are optimized to extract only the information that is maximally correlated with the actual corrosion state. Appropriate performance metrics, such as correlation coefficients, Fisher's Discriminant Ratio (FDR), et al. can be utilized to assist in the selection and validation processes. Figure 5 shows the overall data mining scheme. Image pre-processing, feature extraction and selection are highlighted leading to their utility in pitting corrosion detection, localization, interpretation, and eventually prediction of corrosion states.

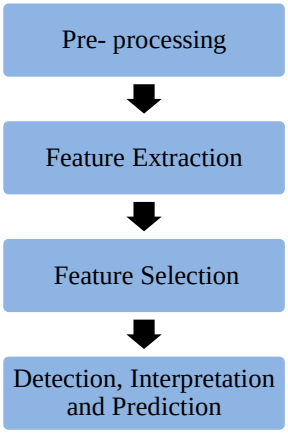


Figure 5. Corrosion data mining scheme.

4.1. Image Pre-processing

Image/data pre-processing involves filtering and preparing the data for further processing. Figure 6 shows a typical sequence of pre-processing steps of corrosion images from surface metrology testing. In the first step, de-noising, discrete stationary wavelet transform (SWT) is applied, and then histogram equalization is performed for contrast enhancement followed by applying a threshold to identify the regions of interest in the image. In this framework, image processing techniques are utilized to pre-process the global panel images as well as the local pitting area images, in preparation for the feature extraction step introduced in Section 4.2. First, globally, for each panel, successive 2D microscopic images were taken and stitched together to obtain the entire panel image. In the whole panel image pre-processing, the rivet-hole areas and artifacts (e.g., stencil-stamp marked numbers) were manually whitened so they would not be confused with corroded regions. Then, in order to identify the pitting corrosion attacked areas, a 2D median filter was applied followed by thresholding (with a threshold of 0.2) to obtain at a binary image. Second, locally, each suspected pitting area was identified from the whole panel image, and a closer microscopy examination was conducted. An example of a local pit identification process is as shown in Figure 7. To identify the pit(s) from the background, the area of each object (i.e., a black region representing a corroded region) in the binary image was calculated. The sum of objects with the area larger than 50 pixels was defined as the total area of the pitting corroded regions. Note that the identification threshold of 50 pixels was set to avoid mistaking dark regions caused by the grain boundaries as pits.

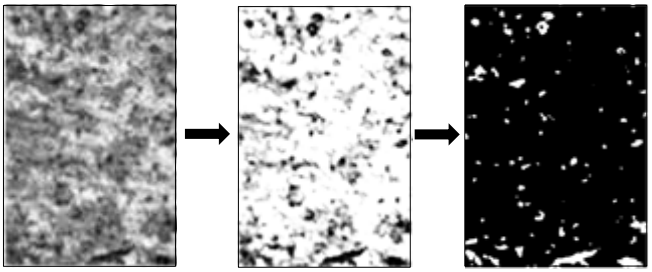
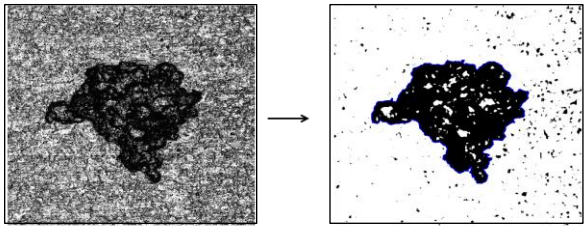


Figure 7. Local pit identification via image processing. Left: Original localized pit image; Right: Pit identified from the



background with the pit edge (in blue) identified by image processing algorithm.

4.2. Feature Extraction

There are several characterization features to quantify the pitting corrosion attack, e.g. corroded area percentage, average pit depth measurement, maximum pit depth measurement, pitting density (pits/mm^2), and remaining wall thickness due to pitting. In addition, image processing techniques can be used to extract morphological and texture features to facilitate pitting corrosion interpretation. The following outlines the features extracted from 2D corrosion images and 3D characterization data, which may facilitate the corrosion detection and interpretation:

1) Corroded Area Percentage

The pitting corroded area percentage is calculated as

$$percent\ area = 100\% \cdot \left(\frac{A_c}{A_I - A_R}\right) \quad (1)$$

where A_c is the area of the corroded region, A_I is the area of the image and A_R is the area of the rivets.

2) Imaging Texture Features using Gray Level Co-occurrence Matrix

2D imaging texture features such as contrast, correlation, energy and homogeneity, as expressed in Eqs. (2-5), are calculated using the normalized gray level co-occurrence matrix (GLCM) denoted as p(i, j). The (i, j) value of the GLCM of an image I has the value of how often a pixel with value i occurs horizontally adjacent to a pixel with value j in image I. The contrast as in Eq. (2) returns a measure of the intensity contrast between a pixel and its neighbor over the whole image. For a constant image, the contrast is 0. The correlation as in Eq. (3) returns a measure ranging between -1 and 1 represents how correlated a pixel is to its neighbor over the whole image. The energy as in Eq. (4) is calculated as the sum of the squared elements in the GLCM. For a constant image, the energy is 1. The homogeneity as in Eq. (5) is a measure of the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

$$contrast = \sum_{i,j} |i-j|^2 p(i,j) \quad (2)$$

$$correlation = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i \sigma_j} \quad (3)$$

$$energy = \sum_{i,j} p(i,j)^2 \quad (4)$$

$$homogeniety = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \quad (5)$$

3) Morphological Features

Morphological features can be extracted from 2D pitting images to characterize the shape of the pitting attacked surface area. Features such as roundness, solidity, eccentricity, major axis length and minor axis length are calculated as expressed in Eqs. (6-10):

$$roundness = \frac{4\pi A}{p^2} \quad (6)$$

where A is the area of the region and p is the perimeter of the region.

$$solidity = \frac{Area}{ConvexArea} \quad (7)$$

where ConvexArea is the area of the convex hull of the region.

For an ellipse defined by $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$, the eccentricity, major axis length and minor axis length are calculated as

$$eccentricity = \sqrt{1 - \frac{b^2}{a^2}} \quad (8)$$

$$L_{Major} = \max(2a, 2b) \quad (9)$$

$$L_{Minor} = \min(2a, 2b). \quad (10)$$

4) Surface Roughness

Surface roughness is a measure of the texture of a surface. It is quantified by the vertical deviations Z(x,y) of a real surface from its ideal form. If these deviations are large, the surface is rough; if they are small the surface is smooth. Roughness is typically considered to be the high frequency, short wavelength component of a measured surface. The 3D surface roughness features are listed in Table 1.

Table 1. Surface roughness parameters and their expressions.

Name	Symbol	Equation
Maximum Height	S_Z	$S_Z = S_P + S_V$
Maximum Peak Height	S_P	$S_p = \max(Z(x,y))$
Maximum Valley Depth	S_v	$S_V = \min(Z(x,y))$
Arithmetic Mean Height	S_a	$S_a = \frac{1}{A}\iint \lvert Z(x,y)\rvert dxdy$
Root Mean Squared Height	S_q	$S_q = \sqrt{\frac{1}{A}\iint \lvert Z(x,y)\rvert^2 dxdy}$
Skewness	S_{sk}	$S_{sk} = \frac{1}{S_q^3}\frac{1}{A}\iint Z(x,y)^3 dxdy$
Kurtosis	S_{ku}	$S_{ku} = \frac{1}{S_q^4}\frac{1}{A}\iint \lvert Z(x,y)\rvert^4 dxdy$

5) Other Characterization Features

Other pit characterization features include the corroded area geometric features (e.g., surface area, circumference), 2D pit profile (line) features (e.g., pit width, pit depth, pit profile cross-sectional area), 3D pit profile features (e.g., pit volume), et al.

4.3. Feature Selection via Performance Metrics

After a sufficient number of image/characterization features are extracted, feature selection can be conducted to determine the smallest subset of features that satisfies given performance criteria. Performance metrics such as correlation coefficient and Fisher discriminant ratio (FDR) can be applied to assess the feature quality. Optimization and Principle Component Analysis (PCA) tools can be used for this purpose. Then a list of "best" features can be selected based on the feature performance. Here we use correlation coefficient and FDR to gauge the image features:

1) Correlation Coefficient

The correlation coefficient is defined as

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (11)$$

where, X and Y are two random variables with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y. The estimate of the correlation coefficient can be expressed as

$$r_{xy} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2 \sum(y_i-\bar{y})^2} \quad (12)$$

where $\bar{x}$ and $\bar{y}$ are the sample means of X and Y.

2) Fisher Discriminant Ratio (FDR)

Fisher's linear discriminant is a classification method that projects high-dimensional data onto a line and performs classification in this one-dimensional space. The projection maximizes the distance between the means of the two classes while minimizing the variance within each class. This defines the Fisher criterion, or FDR, which is maximized over all linear projections. The FDR of two classes is given as

$$FDR = \frac{(\mu_1-\mu_2)^2}{{\sigma_1}^2+{\sigma_2}^2} \quad (13)$$

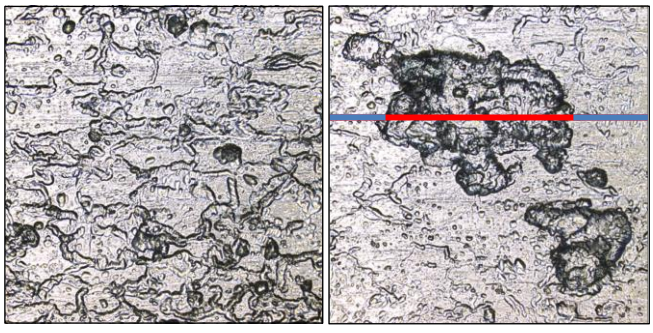
where μ represents a mean, σ represents a variance, and the subscripts denote the two classes.

5. RESULTS AND DISCUSSION

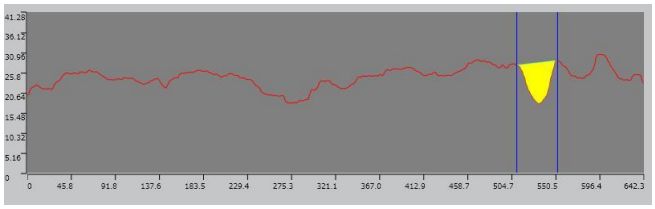
In this paper, we assume that in the accelerated corrosion testing, the corrosion protection coating prevents the corrosion attack up to the maximum hours of corrosion exposure (i.e., 286 hours), and thus we use the measurement from panel coated sides as "baselines", and compare to the one from the panel uncoated sides.

5.1 Corrosion Characterization Features

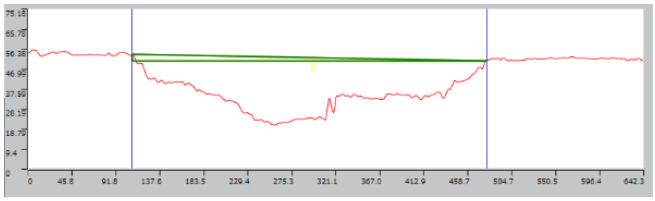
Preliminary global inspection through the profilometer 3D map scan indicated that the corroded panels were pretty flat without noticeable low-frequency surface irregularities, and thus the surface features can be mostly captured by roughness. Therefore, we can omit waviness for this application. Thus, smoothness and spike removal filters were generally applied at the raw profile measurement from the profilometer and the microscope. Figure 8 (a) and (b) provide the 2D microscopic images of the local pitted panel areas of the same size and magnification in Panel 1 and 2, and Figure 8 (c) and (d) illustrate typical pit cross-sectional profiles from Panel 1 and 2 respectively, with (d) corresponding to the colored line marked in (b). Figure 9 shows a 3D topology image of an area of connected pitting in Panel 2. Table 2 lists the 2D pit profile measurement of the colored lines in Figure 8 (b) and Figure 9, of which the pit height represents the maximum pit depth.



(a) (b)



(c)



(d)

Figure 8. 2D characterization of pitted panel areas (642 × 644 μm^2) on the uncoated side of (a) Panel 1, and (b) Panel 2; pit cross-sectional profile measurement (in μm) of (c) a general pit in Panel 1 (with the highlighted cross-sectional area of 240.43 μm^2), and (d) the colored line in (b), Panel 2.

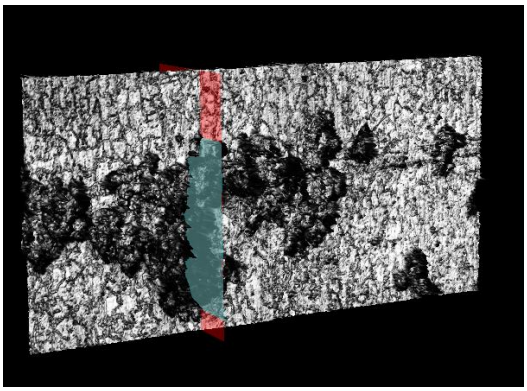


Figure 9. 3D characterization of a pitted panel area (2561 × 1278 μm^2) on the uncoated side of Panel 2, with the corresponding cross-sectional profile measurement as listed in Table 2.

Table 2. Corresponding 2D pit profile measurement (in μm) of the colored lines in Figure 8(b) and Figure 9, Panel 2.

Measurement	**Figure 8 (b)**	**Figure 9**
Width (μm)	369.432	848.483
Height(μm)	3.164	19.895
Length(μm)	369.445	848.717

Except for the 2D pit profile features such as pit width and pit depth, geometric features such as pitting surface area and circumference (as shown in Figure 11 and Table 4) and pit volume (as shown in Figure 10 and Table 3) can also provide solid measures for local pitting severity, of which pit volume is of importance, due to the irregular growth pattern of pitting corrosion. In Figure 10 and Figure 11, a surface height threshold was manually chosen respectively, in order to calculate the corroded surface area and the underneath pitting volume. In Figure 11, as calculated from Table 4, the pitting affected surface area was in total of 258,380.787 μm^2, or 3.94% of the entire examined surface area.

Detailed analysis of the above pitting characterization results revealed some interesting findings. First, morphological analysis of the pits in Panel 1 and Panel 2 indicated that, the nucleated pits, as those general non-visible ones in Panel 1, usually took regular morphological forms, such as hemi-spherical, near-hemispherical and near-conical shapes as indicated in Figure 8 (a) and (c). As the corrosion exposure time increased, a few nucleated pits evolved into irregular shapes with the pit dimension increased, as indicated in Figure 8 (b) and (d). From a side-by-side comparison in Figure 8 (a) and (b), it is noted that, in Panel 2, even though some nucleated pits evolved into bigger and irregular pits, the majority of the pit population were still in a regular shape with similar dimensions as the nucleated pits in Panel 1. Second, as noted from Table 2, a prevalent phenomenon among the big visible pits in Panel 2 and 3 was that, a pit's width was usually significantly larger than its depth, which suggests that the metal dissolution rate was higher at the pit wall than at the pit bottom. In summary, from localized pitting characterization analysis of all three panels, it is concluded that on Panel 1, a number of nucleated pits formed, but generally few big visible pits existed; from Panel 1 to 2, as the corrosion exposure time increased from 133 hours to 209 hours, there emerged a few visible pits assuming irregular shapes, very likely with a much bigger width than depth; from Panel 2 to 3, as exposure time further increased to 286 hours, more and more large visible pits formed, located most likely close to panel edges, rivet hole edges and surface irregularies. Note that, due to the nature of the accelerated corrosion testing, three panels, instead of one, were exposed to three different corrosion emersion times respectively. Thus, an individual pit characterization growth cannot be observed in this study. Instead, 3D microscopic characterization studies of a number of random pits were conducted in each panel. It is indicated from the results of the three panels that, even though there was a big scatter of the characterization data of the visible pits on Panel 2 and 3, the number of big visible pits and the connected pitting areas increased with exposure time.

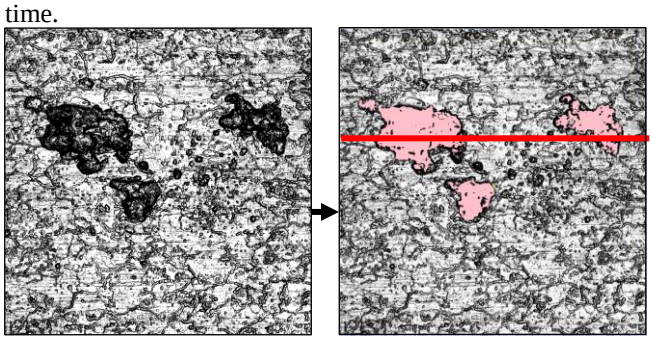


Figure 10. Surface height thresholding procedure to obtain the 3D pitting characterization as shown in Table 3 for a

pitted panel area (1278 ×1281 μm^2) on the uncoated side of Panel 2.

Table 3. Corresponding 3D pitting characterization measurement of the area in Figure 10.

Cross-sectional Area(μm^2) (of the red line in Figure 10)	103,366.090
Surface Area (μm^2)	192,043.495
Volume (μm^3)	1,101,417.185

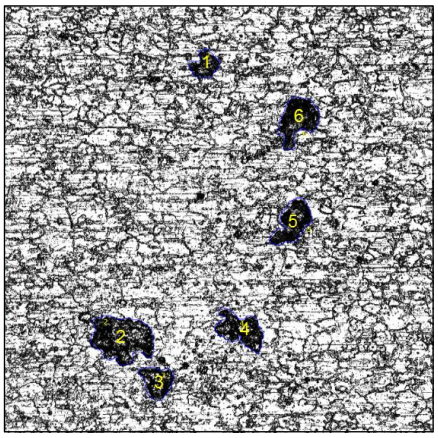


Figure 11. Pitted panel area (2553 ×2568 μm^2) on the uncoated side of Panel 2, with the corresponding 6-pit geometric measurement as listed in Table 4.

Table 4. Corresponding 3D pitting characterization measurement of the area in Figure 11.

No.	Surface Area(μm^2)	Circumference (μm)
1	20,081.765	679.027
2	79,576.806	1,333.879
3	28,428.326	770.822
4	43,645.952	1,216.175
5	39,969.714	1,053.796
6	46,678.224	1,110.563

5.2 Corrosion Image Features

5.2.1 Image Pre-processing

In addition to local pitting characterization analysis, 2D panel images were acquired successively and pre-processed in preparation for corrosion image feature extraction. For each panel, 2D microscopic images of size 37 x 37 mm were taken using LEXT OLS4000 with a magnification setting of 108x, and then stitched together to obtain the entire panel image. Figure 12 depicts the stitched whole panel microscopic images of Panel 1, 2 and 3 and their corresponding binary images after image pre-processing.

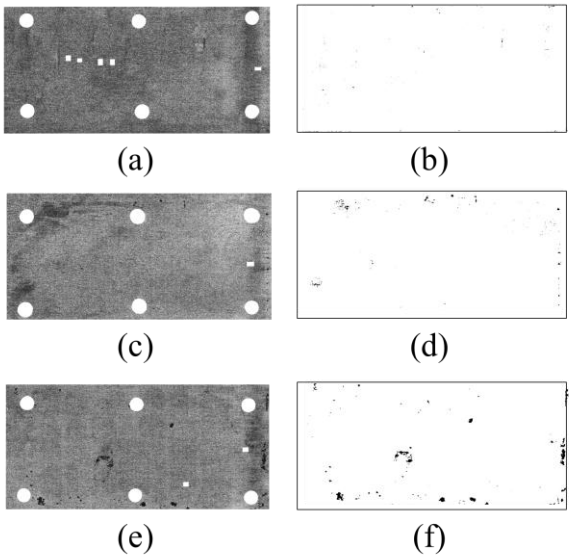


Figure 12. Whole panel image pre-processing. Left column: intermediary images with rivet holes and marked numbers whitened of (a) Panel 1 with 133-hr corrosion exposure, (c) Panel 2 with 209-hr corrosion exposure, (e) Panel 3 with 286-hr corrosion exposure. Right column: binary images after pre-processing of (b) Panel 1, (d) Panel 2, (f) Panel 3.

5.2.2 Feature Extraction, Selection and Data Mining

Features extracted from segments of the corrosion images can be used to classify the state of corrosion in the corresponding image segment. Figure 13 shows an example set of corrosion images used for feature extraction. The top row is a set of 8 low corrosion images and the bottom row is a set of 8 high corrosion images. Contrast, correlation, energy and homogeneity features of the example corrosion images in Figure 13 were calculated and illustrated in Figure 14. The corresponding feature performance was evaluated using FDR as listed in **Table 5**. Table 5 indicates that correlation, energy and homogeneity are good image features for corrosion detection and corrosion state classification, whereas contrast performs poorly.

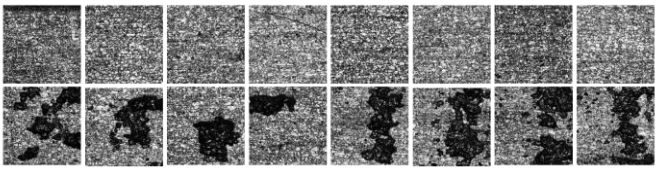


Figure 13. Example corrosion images. Top row: low corrosion. Bottom row: high corrosion.

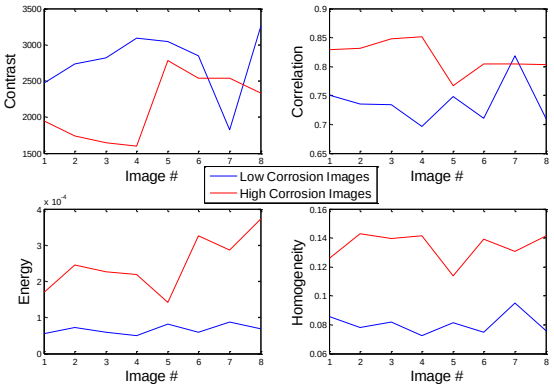


Figure 14. Contrast, Correlation, Energy and Homogeneity features of low and high corrosion images from Figure 13 (image number ascends correspond to the sequence from left to right in each row of Figure 13).

Table 5. FDR values of image features.

Features	Contrast	Correlation	Energy	Homogeneity
FDR	0.9604	2.2084	95.1962	27.3738

Figure 15 shows the corroded area percentage of the panels that had corrosion exposure times of 133, 209 and 286 hours. The resulting corroded area percentage feature was highly correlated with the measured panel mass loss as shown in Figure 15. The correlation coefficient r_{xy} of the corroded area percentage and the corresponding measured panel mass loss is 0.9727.

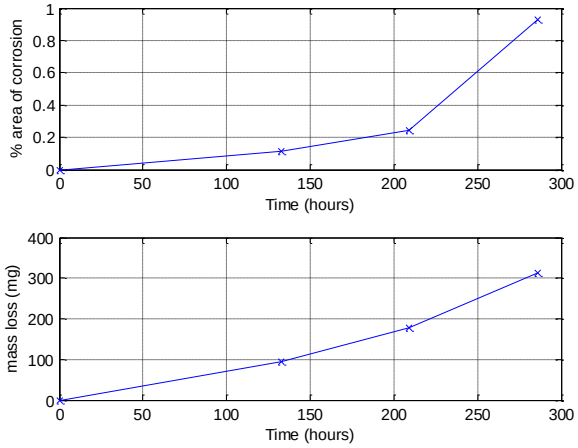


Figure 15. Top: Corroded area percentage over time. Bottom: Measured mass loss (mg) over time.

6. CONCLUSIONS

This paper reports results from an experimental investigation of pitting corrosion detection and interpretation on aluminum alloy panels using surface metrology methods, image processing and information processing techniques. Accelerated corrosion testing of the lap-joint panels was performed in a cyclic corrosion chamber running ASTM G85-A5 salt fog test. Then the global and local corrosion behaviors were imaged and characterized via microscopy and profilometry examination. Data mining techniques are utilized, including image pre-processing, image and characterization feature extraction and selection, to facilitate the study of corrosion morphological behavior and its progression as a function of corrosion exposure time. The morphological study showed that facing electrochemical corrosion attack, pits initiated and predominantly assumed in regular shapes, but underwent irregular thus progressive geometric transitions associated with increased corrosion exposure time. This study also examined a list of promising characterization and image features and conducted the performance evaluation of some representative features for corrosion interpretation. This study is a first step in the process of understanding, assessing and responding to the pitting corrosion and ultimately preventing material failure to insure aircraft structural integrity. Future work may include more rigorous testing and analysis methods, e.g., to study an individual pit evolution over time, and the evolution from pitting to cracking under stress corrosion condition; and further in the direction of aircraft structure health management, to accurately model the corrosion progression, assess the corrosion states, and predict the corrosion-induced structure failure.

ACKNOWLEDGEMENT

We acknowledge the support for this program from BAA/RIF contract # FA8650-12-C-0001 monitored by Mr. Feraidoon Zahiri.

REFERENCES

Belanger, P., Cawley, P. and Simonetti, F. (2010). Guided wave diffraction tomography within the Born approximation, *IEEE Trans UFFC*, Vol. 57, pp. 1405-1418.

Clark, T., (2009). *Guided wave health monitoring of complex structures*. Doctoral dissertation. Imperial College London, London, United Kingdom.

Frankel, G. S. (1998). Pitting corrosion of metals a review of the critical factors. *Journal of the Electrochemical Society*, 145(6), 2186-2198.

Huang, T.-S. & Frankel, G. S. (2006). Influence of Grain Struc- ture on Anisotropic Localized Corrosion Kinetics of AA7xxx-T6 Alloys. *Corrosion Engineering, Science and Technology*, Vol. 41, No. 3, pp. 192-199. doi:10.1179/174327806X120739.

Li, H., Michaels, J. E., Lee, S. J., Michaels, & T. E., Thompson, D. O., & Chimenti, D. E. (2012).

Quantification of surface wetting in plate-like structures via guided waves. *In AIP Conference Proceedings-American Institute of Physics*, Vol. 1430, No. 1, p. 217.

Pereira, M. C., Silva, J. W., Acciari, H. A., Codaro, E. N., & Hein, L. R. (2012). Morphology Characterization and Kinetics Evaluation of Pitting Corrosion of Commercially Pure Aluminium by Digital Image Analysis. *Materials Sciences & Applications*, 3(5), pp. 287-293.

Rao, K. S., & Rao, K. P. (2004). Pitting Corrosion of Heat-Treatable Aluminium Alloys and Welds: A Review. *Transactions of the Indian Institute of Metals*, Vol. 57, No. 6, pp. 593-610.

Shell, E. B., Buchheit, R. G., & Zoofan, B. (2005). Correlation of residual fatigue life with quantified NDE measurements. *International Journal of Fatigue*, 27(2), 105-112.

Silva, J. W. J., Bustamante, A. G., Codaro, E. N., Nakazato, R. Z., & Hein, L. R. O. (2004). Morphological Analysis of Pits Formed on Al 2024-T3 in Chloride Aqueous Solution. *Applied Surface Science*, Vol. 236, No. 1-4, pp. 356-365. doi:10.1016/j.apsusc.2004.05.007.

Szklarska-Smialowska, Z. (1999). Pitting corrosion of aluminum. *Corrosion Science*, 41(9), 1743-1767.

Trdan, U., Ocaña, J.L., Grum, J. (2009). Surface Evaluation of Laser Shock Processed Aluminum Alloy After Pitting Corrosion Attack with Optical 3-D Metrology Method. *The 10th International Conference of the Slovenian Society for Non-Destructive Testing*, September 1-3, 2009, Ljubljana, Slovenia.

BIOGRAPHIES

Honglei Li received her masters' degree in Electrical and Computer Engineering (ECE) from the Georgia Institute of Technology, and in Instrumental Engineering from Shanghai Jiao Tong University in 2010. Prior to joining Intelligent Control Systems Laboratory (ICSL) in 2012 as a graduate research assistant, she had years of interdisciplinary research experience in Non-destructive Testing & Evaluation, and biomedical engineering fields. Currently she is also working on her doctoral degree in ECE at the Georgia Institute of Technology. Her current research interests include methodologies for Structural Health Monitoring (SHM), Prognostics & Health Management (PHM), as well as incorporating of PHM for engineered system life cycle management.

Jiaming Li received the B.S degree from the University of Science and Technology of China, in 2011, the M.S. degree from Georgia Institute of Technology, in 2014. Now he is working on his Ph.D. degree in Georgia Institute of Technology. He is currently a Graduate Research Assistant with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His current research interests include human-machine interface, fault diagnosis and failure prognosis.

Javier Echauz is Senior Research/Data Scientist and President of JE Research, Inc., a biomedical engineering consultancy he founded in 2008 focused on computational intelligence and devices for epilepsy. In 1995, he obtained a Ph.D. degree from Georgia Institute of Technology on intelligent controls and EEG analysis under mentorship of Dr. George Vachtsevanos at Tech, and jointly Dr. Kimford Meador at Medical College of Georgia. Dr. Echauz was a tenured Associate Professor and Associate Director of the Electrical and Computer Engineering Department at University of Puerto Rico—Mayagüez before returning to Atlanta in 2000. With the Jacoby Group support, he cofounded IntelliMedix, Inc., and later BioQuantix Corp., where algorithms and intellectual property were developed and crosslicensed in collaboration with Dr. Brian Litt at Emory and Penn. He was Senior Research Scientist at NeuroPace, Inc., responsible for the seizure detection system of an implantable brain neurostimulator for epilepsy, recenty approved by the FDA. With NeuroVista Corp., Dr. Echauz developed algorithms and data quality protocols resulting in an implantable seizure advisory system. He later developed text mining interactive web applications for drug discovery, and built an EEG system for monitoring level of consciousness during seizures and PNES. Dr. Echauz has co-authored over 60 technical papers, 5 book chapters, and 12 issued patents.

Douglas W. Brown is the Senior Systems Engineer for Analatom, Inc. He received the bachelor of science degree in electrical engineering from the Rochester Institute of Technology and his master of science and doctor of philosophy degrees in electrical engineering from the Georgia Institute of Technology. Dr. Brown has ten years of experience developing and maturing Prognostics & Health Management (PHM) and fault-tolerant control systems in avionics application. He is a recipient of the National Defense Science and Engineering Graduate (NDSEG)

Fellowship and has received several best-paper awards in his work in PHM and fault-tolerant control.

George J. Vachtsevanos is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, a M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at Georgia Tech where faculty and students are conducting research in intelligent control, neurotechnology and cardiotechnology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 240 technical papers and is a senior member of IEEE. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.

A Novel Linear Polarization Resistance Corrosion Sensing Methodology for Aircraft Structure

Douglas W. Brown[1], Richard J. Connolly[2], Bernard Laskowski[3], Margaret Garvan[4], Honglei Li[5], Vinod S. Agarwala[6], and George Vachtsevanos[7]

[1,2,3] *Analatom, Inc., 3210 Scott Blvd., Santa Clara, CA 95054, USA*
doug.brown@analatom.com
richard.connolly@analatom.com
bernard.laskowski@analatom.com

[4,5,7] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
mgarvan3@gatech.edu
honglei.li@gatech.edu
gjv@gatech.edu

[6] *Iron Pillar Technical Services, 1600 Green Street, Philadelphia, PA 19130, USA*
vsagarwala@comcast.net

ABSTRACT

A direct method of measuring corrosion on a structure using a micro-linear polarization resistance (μLPR) sensor is presented. The new three-electrode μLPR sensor design presented in this paper improves on existing LPR sensor technology by using the structure as part of the sensor system, allowing the sensor electrodes to be made from a corrosion resistant or inert metal. This is in contrast to a two-electrode μLPR sensor where the electrodes are made from the same material as the structure. A controlled experiment, conducted using an ASTM B117 salt fog, demonstrated the three-electrode μLPR sensors have a longer lifetime and better performance when compared to the two-electrode μLPR sensors. Following this evaluation, a controlled experiment using the ASTM G85 Annex 5 standard was performed to evaluate the accuracy and precision of the three-electrode μLPR sensor when placed between lap joint specimens made from AA7075-T6. The corrosion computed from the μLPR sensors agreed with the coupon mass loss to within a 95% confidence interval. Following the experiment, the surface morphology of each lap joint was determined using laser microscopy and stylus-based profilometry to obtain local and global surface images of the test panels. Image processing, feature extraction, and selection tools were then employed to identify the corrosion mechanism (e.g. pitting, intergranular).

Douglas Brown et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Recent studies have exposed the generally poor state of our nation's critical infrastructure that has resulted from wear and tear under excessive operational loads and environmental conditions. The British Standards Institution's Publicly Available Specification for the optimized management of physical assets defines asset management as the "systematic and coordinated activities and practices through which an organization optimally and sustainably manages its assets and asset systems, their associated performance, risks and expenditures over their life cycles for the purpose of achieving its organizational strategic plan." The motivation for effective asset management is driven by owners' desire for higher value assets at less overall costs, thus extracting the maximum value from their assets (Herder & Wijnia, 2011). Condition-based maintenance aims to maximize asset value by extending the useful life of assets through mitigation of unnecessary maintenance actions performed during schedule-based maintenance strategies (Huston, 2010). By providing maintenance engineers with information regarding the health of the structure, maintenance can be performed on a basis of necessity unique to each asset, as opposed to schedule-based predictions formed on statistical trends of operational reliability. These systems must be low-cost and simple to install with a user interface designed to be easy to operate.

To reduce the cost and complexity of such a system for monitoring corrosion in an avionics environment, a generic interface node using low-powered wireless communications has

Figure 1. AN110 installed on a C-130H

been developed. This node can communicate with a myriad of common sensors used in SHM. In this manner a structure such as a bridge, aircraft, or ship can be fitted with sensors in any desired or designated location and format without the need for communications and power lines that are inherently expensive and complex to route. Data from these nodes is transmitted to a central communications personal computer for data analysis. An example of this is provided in Figure 1 showing an embedded AN110 SHM system installed on a C-130H aircraft.

The micro-linear polarization resistance (μLPR) sensor presented in this paper improves on existing LPR technology by using the structure as part of the sensing system. The sensor includes three electrodes, where each electrode is fabricated on a flexible substrate to create a circuit consisting of gold-plated copper. The first two electrodes, or the counter and reference electrodes, are configured in an interdigitated fashion with a separation distance of 8mil. The flex cable contains a porous membrane between the pair of electrodes and the structure. A third electrode, or the working electrode makes electrical contact to the structure through a 1mil thick electrically conductive transfer tape placed between the electrode and structure. The reference and counter electrodes are electrically isolated from the working electrode and physically separated from the surface of the structure by 1mil. The flex cable can be attached to the structure with adhesives or in the case of placement in a butt joint or lap joint configuration, by the mechanical forces present in the joint itself. Corrosion is computed from known physical constants, by measuring the polarization resistance between the electrolytic solution and the structure. Further improvements are realized by narrowing the separation distance between electrodes, which minimizes the effects due to solution resistance. This enables the μLPR to operate more effectively outside a controlled aqueous environment, such as an electrochemical cell, in a broad range of applications (eg. civil engineering, aerospace, petrochemical).

The remainder of the paper is organized as follows. Section 2 provides background information on different corrosion sensing technologies, LPR theory, and the new 3-electrode μLPR sensor design. Section 3 describes the experimental procedure used to evaluate the new sensor design through a controlled ASTM G85 Annex 5 cyclic salt fog test. Section 4 presents the results of experimental testing comparing the corrosion rate computed from μLPR sensor data with measured mass loss. Also presented are correlations between features, exposure time, and μLPR sensor measurements. Finally, the paper is concluded in Section 5 with a summary of the findings and future work.

2. BACKGROUND

Corrosion sensors can be distinguished by the following categories, *direct* or *indirect* and *intrusive* or *non-intrusive*. Direct corrosion monitoring measures a response signal, such as a current or potential, resulting from corrosion. Examples of common direct corrosion monitoring techniques are: corrosion coupons, electrical resistance (ER), electrochemical impedance spectroscopy (EIS), and linear polarization resistance (LPR) techniques. Whereas, indirect corrosion monitoring techniques measure an outcome of the corrosion process. Two of the most common indirect techniques are ultrasonic testing and radiography. An intrusive measurement requires access to the structure. Corrosion coupons, ER, EIS, and LPR probes are intrusive since they have to access the structure. Non-intrusive techniques include ultrasonic testing and radiography.

Each of these methods have advantages and disadvantages. Corrosion coupons provide the most reliable physical evidence possible. Unfortunately, coupons usually require significant time in terms of labor and provide time averaged data that can not be utilized for real-time or on-line corrosion monitoring (Harris, Mishon, & Hebbron, 2006). ER probes provide a basic measurement of metal loss, but unlike coupons, the value of metal loss can be measured at any time, as frequently as required, while the probe is *in situ* and permanently exposed to the structure. The disadvantage is ER probes require calibration with material properties of the structure to be monitored. The advantage of the LPR technique is that the measurement of corrosion rate is made instantaneously. This is a more powerful tool than either coupons or ER where the fundamental measurement is metal loss and some period of exposure is required to determine corrosion rate. The disadvantage to the LPR technique is that it can only be suc-

cessfully performed in relatively clean aqueous electrolytic environments (*Introduction to Corrosion Monitoring*, 2012). EIS is a very powerful technique that can provide a corrosion rate and classification of the corrosion mechanism. EIS measures the magnitude and phase response of an electrochemical cell. Physical parameters, such as the polarization resistance, solution resistance, and double-layer capacitance, can be derived from these responses, which provides more information than just LPR alone. The disadvantage with EIS is that it uses sophisticated instrumentation that requires a controlled setting to obtain an accurate spectrum. In fielded environments, EIS is highly susceptible to noise. Additionally, interpretation of the data can be difficult (Buchheit, Hinkebein, Maestas, & Montes, 1998). Ultrasonic testing and radiography can be used to detect and measure (depth) corrosion through non-destructive and non-intrusive means (Twomey, 1997). The disadvantage with the ultrasonic testing and radiography equipment is the same with corrosion coupons, both require significant time in terms of labor and can not be utilized for real-time or on-line corrosion monitoring. As this paper is focused on a three-electrode μLPR sensor, the remainder of the background will focus on LPR.

2.1. LPR Theory

Corrosion occurs as a result of oxidation and reduction reactions occurring at the interface of a metal and an electrolyte solution. This process occurs by electrochemical half-reactions; (1) anodic (oxidation) reactions involving dissolution of metals in the electrolyte and release of electrons, and (2) cathodic (reduction) reactions involving gain of electrons by the electrolyte species like atmospheric oxygen, O_2, H_2O, or H^+ ions in an acid (Harris et al., 2006). The flow of electrons from the anodic reaction sites to the cathodic reaction sites creates a corrosion current. The electrochemically generated corrosion current can be very small (on the order of nanoamperes) and difficult to measure directly. Application of an external potential exponentially increases the anodic and cathodic currents, which allows instantaneous corrosion rates to be extracted from the polarization curve. Extrapolation of these polarization curves to their linear region provides an indirect measure of the corrosion current, which is then used to calculate the rate of corrosion (Burstein, 2005).

The electrochemical technique of LPR is used to study corrosion processes since the corrosion reactions are electrochemical reactions occurring on the metal surface. Modern corrosion studies are based on the concept of mixed potential theory postulated by Wagner and Traud, which states that the net corrosion reaction is the sum of independently occurring oxidation and reduction reactions (Wagner & Traud, 1938). For the case of metallic corrosion in presence of an aqueous medium, the corrosion process can be written as,

$$\mathrm{M} + z\mathrm{H_2O} \underset{b}{\overset{f}{\leftrightarrow}} \mathrm{M}^{z+} + \frac{z}{2}\mathrm{H_2} + z\mathrm{OH}^-, \tag{1}$$

where z is the number of electrons lost per atom of the metal. This reaction is the result of an anodic (oxidation) reaction,

$$\mathrm{M} \underset{b}{\overset{f}{\leftrightarrow}} \mathrm{M}^{z+} + ze^-, \tag{2}$$

and a cathodic (reduction) reaction,

$$z\mathrm{H_2O} + ze^- \underset{b}{\overset{f}{\leftrightarrow}} \frac{z}{2}\mathrm{H_2} + z\mathrm{OH}^-. \tag{3}$$

It is assumed that the anodic and cathodic reactions occur at a number of sites on a metal surface and that these sites change in a dynamic statistical distribution with respect to location and time (Kossowsky, 1989). Thus, during corrosion of a metal surface, metal ions are formed at anodic sites with the loss of electrons and these electrons are then consumed by water molecules to form hydrogen molecules. The interaction between the anodic and cathodic sites as described on the basis of mixed potential theory is represented by well-known relationships using current (reaction rate) and potential (driving force). For the above pair of electrochemical reactions (2) and (3), the relationship between the applied current I_a and applied potential, E_a, follows the Butler-Volmer equation,

$$I_a = I_{corr}\left[e^{2.303(E_a - E_{corr})/\beta_a} - e^{-2.303(E_a - E_{corr})/\beta_c}\right], \tag{4}$$

where β_a and β_c are the anodic and cathodic Tafel parameters given by the slopes of the polarization curves $\partial E_a / \partial \log_{10} I_a$ in the anodic and cathodic Tafel regimes, respectively and E_{corr} is the corrosion, or open circuit potential (Bockris, Reddy, & Gambola-Aldeco, 2000). The corrosion current, I_{corr}, cannot be measured directly. However, *a priori* knowledge of β_a and β_c along with a small signal analysis technique, known as polarization resistance, can be used to indirectly compute I_{corr}. The polarization resistance technique, also referred to as linear polarization, is an experimental electrochemical technique that estimates the small signal changes in I_a when E_a is perturbed by $E_{corr} \pm 10\mathrm{mV}$ (G102, 1994). The slope of the resulting curve over this range is the polarization resistance,

$$R_p \triangleq \left.\frac{\partial E_a}{\partial I_a}\right|_{|E_a - E_{corr}| \le 10\mathrm{mV}}. \tag{5}$$

ASTM standard G59 outlines procedures for measuring polarization resistance. Potentiodynamic, potential step, and current-step methods can be used to compute R_p (G59, 1994). The potentiodynamic sweep method is the most common method for measuring R_p. A potentiodynamic sweep is conducted by applying E_a between $E_{corr} \pm 10\mathrm{mV}$ at a slow scan rate, typically $0.125\,\mathrm{mV/s}$. A linear fit of the resulting E_a vs. I_a curve is used to compute R_p. Note, the applied current, I_a, is the total applied current and is not multiplied by the electrode area so R_p as defined in (5) has units of Ω. Provided that $|E_a - E_{corr}|/\beta_a \ll 1$ and $|E_a - E_{corr}|/\beta_c \ll 1$, the first order

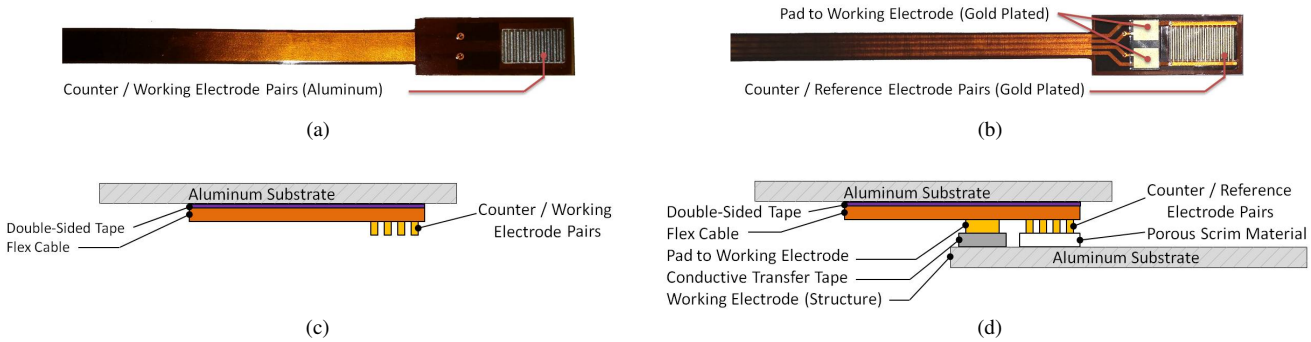


Figure 2. The (a) two-electrode μLPR sensor, (b) three-electrode μLPR sensor, (c) two-electrode μLPR sensor identifying each sensor element when mounted to a substrate, and (d) three-electrode μLPR sensor identifying each sensor element when attached using the structure as the third electrode.

Taylor series expansion $e^x \approx 1+x$ can be applied to (4) and (5) to arrive at the Stern-Geary equation,

$$I_{corr} = \frac{B^\star}{R_p}, \tag{6}$$

where,

$$B^\star = \frac{\beta_a \beta_c}{2.303\,(\beta_a + \beta_c)}. \tag{7}$$

Knowledge of R_p, β_a, and β_c enables direct determination of I_{corr} at any instant in time. The corrosion rate, R_{loss}, can be found by applying Faraday's law,

$$R_{loss}\,(t) = \frac{B_{loss}}{R_p\,(t)}, \tag{8}$$

where,

$$B_{loss} = \frac{B^\star}{F A_{sen}} \left(\frac{AW}{z} \right), \tag{9}$$

such that F is Faraday's constant, z is the number of electrons lost per atom of the metal during an oxidation reaction, A_{sen} is the effective area of the sensor, and AW is atomic weight. The total mass loss, M_{loss}, due to corrosion can be found by integrating (8),

$$M_{loss}\,(t) = \int_{t_0}^{t} R_{loss}\,(\tau)\,d\tau. \tag{10}$$

Finally, since R_p is not measured continuously (10) needs to be discretized for the sample period T_s,

$$M_{loss}\,(t)\Big|_{t=NT_s} = T_s \sum_{k=1}^{N} R_{loss}\,(kT_s)\,. \tag{11}$$

2.2. Sensor Design

The two-electrode μLPR design consists of a sensor with interdigitated electrodes photo-etched from 2 mil aluminum shim-stock material with a thickness and separation distance of 12 mil. In this configuration one of the electrode pairs acts as the counter electrode (cathode) and the other as the working electrode (anode). The sensor is designed to corrode in the same environment as the structure, effectively measuring the corrosivity of the environment. An image of the two-electrode μLPR sensor is provided in Figure 2(a). An illustration showing the two-electrode μLPR sensor mounted to the structure is shown in Figure 2(c).

Improving on the two-electrode design, the three-electrode μLPR is fabricated on a flexible Kapton substrate where each electrode is coated with a noble metal. The first two electrodes, counter and reference electrodes, are fabricated using 0.5 oz. copper with an electroless nickel immersion gold (ENIG) finish and an overall thickness of 1 mil. The counter and reference electrode pair is configured in a interdigitated geometric layout with a separation distance of 9 mil. The flex cable contains an insulating porous scrim material between the pair of electrodes and the structure. A third electrode, made from the same ENIG finish, is placed in close proximity to the counter and reference electrodes; electrical contact is made with the structure by placing a 1 mil thick electrically conductive transfer tape between the electrode and structure. This allows the structure to serve as the working electrode for the sensor measurement. The flex cable, shown in Figures 2(b) and (d), can be attached to the structure through the use of adhesives or in the case of placement in a butt joint or lap joint configuration, the holding force is provided by the joint itself.

3. EXPERIMENTAL PROCEDURES

3.1. Tafel Measurements

ASTM standard G59 outlines the procedure for measuring the Tafel slopes, β_a and β_c. First, E_{corr} is measured from the open circuit potential. Next, E_a is initialized to E_corr-250mV. Then, a potentiodynamic sweep is conducted by increasing E_a from $E_{corr} - 250\text{mV}$ to $E_{corr} + 250\text{mV}$ at a slow

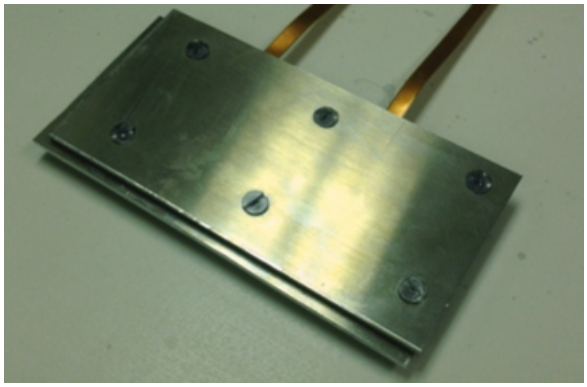


Figure 3. AA7075-T6 lap joint assembly.

Figure 4. Panels shown in the corrosion chamber prior to the experiment.

scan rate, typically $0.125\,\mathrm{mV/s}$. Finally, a Tafel curve is plotted for E_a vs. $\log_{10} I_a$. Values for β_a and β_c are estimated from the slopes of the linear extrapolated anodic and cathodic currents.

3.2. Sample Preparation

Lap joint samples were made using two 6" by 3" panels made from AA7075-T6 with a thickness of 1/8". These panels were secured together with six polycarbonate fasteners. Before assembly of the lap joint each panel was cleaned with a 35 min immersion into a constantly stirred solution of $50\,\mathrm{g/L}$ Turco 4215 NC-LT at 65°C. After completing this alkaline cleaning the panels were rinsed with deionized water and immersed into a 70% solution of nitric acid solution for 5 min at 25°C. The samples were then rinsed again in the deionized water and air dried. Weights were recorded to the nearest fifth significant figure and the samples were stored in a desiccator. Once the panels were prepared and massed, two μLPR sensors were installed between the panels. At this point the six polycarbonate bolts were torqued down evenly to $2\,\mathrm{N\cdot m}$. This lap joint assembly is shown in Figure 3. After assembling the lap joints, the samples were evenly coated with 2 mils of epoxy-based paint and 2 mils of polyurethane on all exposed surfaces. These coatings were allowed to fully seal over a 24 hour period at 35°C before testing.

3.3. Comparing Two vs. Three Electrode Design

A preliminary experiment was performed to highlight the benefits between a two-electrode μLPR sensor made from AA7075-T6 and a three-electrode μLPR sensor made from nickel. This experiment was performed by placing four two-electrode μLPR and four three electrode μLPR sensors into a beaker filled with a B117 salt solution modified to a pH of 5.5. A stirbar was used to constantly mix the solution. The sensors were placed inside the beaker around a plastic cylindrical fixture. The two and three-electrode μLPR sensors were evenly spaced in an alternating arrangement. Approximately every 4 days, the coupons were removed, cleaned, massed and then returned to the beaker to resume the experiment.

3.4. Accelerated Lap Joint Testing

Corrosion tests were performed in a cyclic corrosion chamber running the ASTM G85 Annex 5 test. This test consisted of two one-hour steps. The first step involved exposing the samples to a salt fog for a period of one-hour at 25°C. The electrolyte solution composing the fog was 0.05% sodium chloride and 0.35% ammonium sulfate in deionized water. This step was followed by a dry-off step, where the fog was purged from the chamber while the internal environment was heated to 35°C. Each panel was positioned at a 60° angle with the flex tape facing downward, as not to allow a direct pathway for condensate to travel into the lap joints. Electrical connections for the μLPR sensors were made to an AN110 positioned outside the chamber by passing extension cables through a bulkhead. Temperature, relative humidity, and μLPR data were acquired at 1 min intervals.

3.5. Sample Cleaning

Samples were removed from the environmental chamber and disassembled. Following disassembly, the polyurethane and epoxy coatings on the aluminum panels were removed by placing them in a solution of methyl ethyl ketone. After immersion for 30 min the panels were removed and rinsed with deionized water. These panels were again alkaline cleaned with a 35 min immersion into a constantly stirred solution of $50\,\mathrm{g/L}$ Turco 4215 NC-LT at 65°C. This was followed by a deionized water rinse and immersion into a 90°C solution of 4.25% phosphoric acid containing $20\,\mathrm{g/L}$ chromium trioxide for 10 min. Following the phosphoric acid treatment, panels were rinsed with deionized water and placed into a 70% nitric acid solution for 5 min at 25°C. Panels were then rinsed with deionized water, dipped in ethanol, and dried with a heat gun. This cleaning process was repeated until mass values for the panels stabilized. These values were then compared with mass loss values calculated from the μLPR data.

4. RESULTS

4.1. Comparing Two vs. Three Electrode Design

The Tafel constants were acquired while the panels were undergoing a wetting cycle. The Tafel constants were acquired and plotted as applied voltage vs. the logarithm of applied current magnitude, shown in Figure 5. From this plot the Tafel constants were computed as, $\beta_a = 0.40\,\mathrm{V/dec}$ and $\beta_c = 0.15\,\mathrm{V/dec}$. The corrosion constant, B_{loss}, was computed using (9) with the material properties for AA7075-T6 and sensor properties defined in the nomenclature. Note, the Tafel slope is an intensive parameter and does not depend on the electrode surface area. If the Tafel constants cannot be extrapolated, is not uncommon to approximate β_a and $\beta_c \approx 0.15\,\mathrm{V/dec}$.

The total corrosion for each sensor was computed by applying (10) to integrate the corrosion rate with respect to time. For the first 300 hours of the experiment, both sensors produce comparable results. However, at 300 hours the overall LPR reading began to drop and the variance between sensor readings started to increase, as shown in Figures 6(a) and (b). This may result from a reduction in the effective surface area of the electrodes as a result of the corrosion process. As more corrosion begins to accumulate, the fingers become less and less effective. In contrast, the 95% confidence band for the three-electrode μLPR sensor remained relatively constant throughout the experiment, shown in Figure 6(c) and (d).

4.2. Lap Joint Testing Results

After selecting the three electrode μLPR for further evaluation, a set of four lap joints were assembled. These assemblies were tested over a maximum period of 286 hours, where the environment inside the chamber was cyclically varied in temperature and humidity according to ASTM G85 Annex 5 to promote corrosion. Panels were removed at 133, 209, 286, and 286 hours into the experiment, respectively. Plots of the measured temperature and humidity vs. time are provided in Figure 8. The corrosion rate, shown in Figure 7, was computed from R_p measurements using (8) along with B_{loss} computed during the previous experiment. The total corrosion, shown in Figure 9(a), was computed for each panel by applying (10) to integrate the corrosion rate with respect to time. The error bars correspond to the standard deviation observed at the time when the mass loss was computed. Finally, the measured and computed corrosion from the μLPR measurements were compared in a scatter plot, shown in Figure 9(b). The error bars in the y-direction correspond to observation error. These results indicate the measured corrosion correlated with the computed corrosion to within 95% confidence (two standard deviations of the observation error).

4.3. Lap Joint Imaging Feature

Microscopic images were acquired over a field size of $37\,\mathrm{mm} \times 37\,\mathrm{mm}$ at a magnification of 108x using the LEXT OLS4000 3D Laser Measuring Microscope. Comprehensive images of each panel was created by stitching together adjacent images. The rivet holes and numbers were manually changed to be white so they wouldn't be confused with corroded regions. To get the features a 2D median filter was applied followed by thresholding (using a threshold of 0.2) to get a binary image. The area for each object (each black region is considered to be an object) in the binary image was calculated. The sum of objects with an area larger than 50 pixels (this was to avoid counting dark regions caused by the grain boundaries as pits) was taken to be the area of the corroded region. The percent area of the corrosion was calculated as,

$$P_{area} = 100\% \cdot \frac{A_{corr}}{A_{image} - A_{rivets}}, \tag{12}$$

where A_{corr} is the area of the corroded region, A_{image} is the area of the image, and A_{rivets} is the area of the rivets. Figure 11 shows the original images of each panel along with a binary image for the specimens removed 133 hours, 209 hours and 286 hours into the experiment. Figure 10 shows plots of (a) P_{area} vs. time and (b) P_{area} vs. computed corrosion.

5. CONCLUSION

A new μLPR sensor design was presented for direct corrosion monitoring in structural health management (SHM) applications. The new design improves on existing technologies by: (1) using the structure as part of the sensor measurement; (2) improving sensor lifetime by making the electrodes from a non-corrosive material; and (3) improving on sensor performance by reducing the separation distance between the working, reference, and counter electrodes. Corrosion tests were performed in a cyclic corrosion chamber running ASTM G85-A5 salt fog test. The results indicate the μLPR sensor data correlated with the measured mass loss to within 95% confidence (two standard deviations of the observation error). This demonstrates the μLPR sensor can accurately measure the change in the corrosion rate as a function of time for a given electrolyte condition. Future work includes:

- Demonstrate μLPR sensor accurately measures the corrosion rate as a function of solution conductivity.
- Establish the μLPR sensor can accurately measure corrosion in atmospheric conditions where corrosion rates are lower than in an "accelerated corrosion chamber".
- Investigate the surface morphology of the coupons using a scanning electron microscope (SEM) and correlate the measured corrosion rate as a function of corrosion behavior as determined by the μLPR sensor data over time.

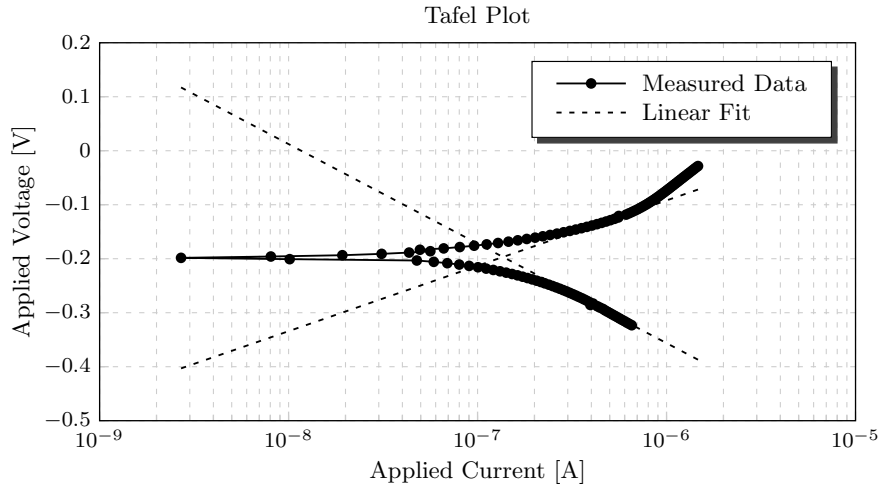


Figure 5. Tafel plot of the μLPR sensors.

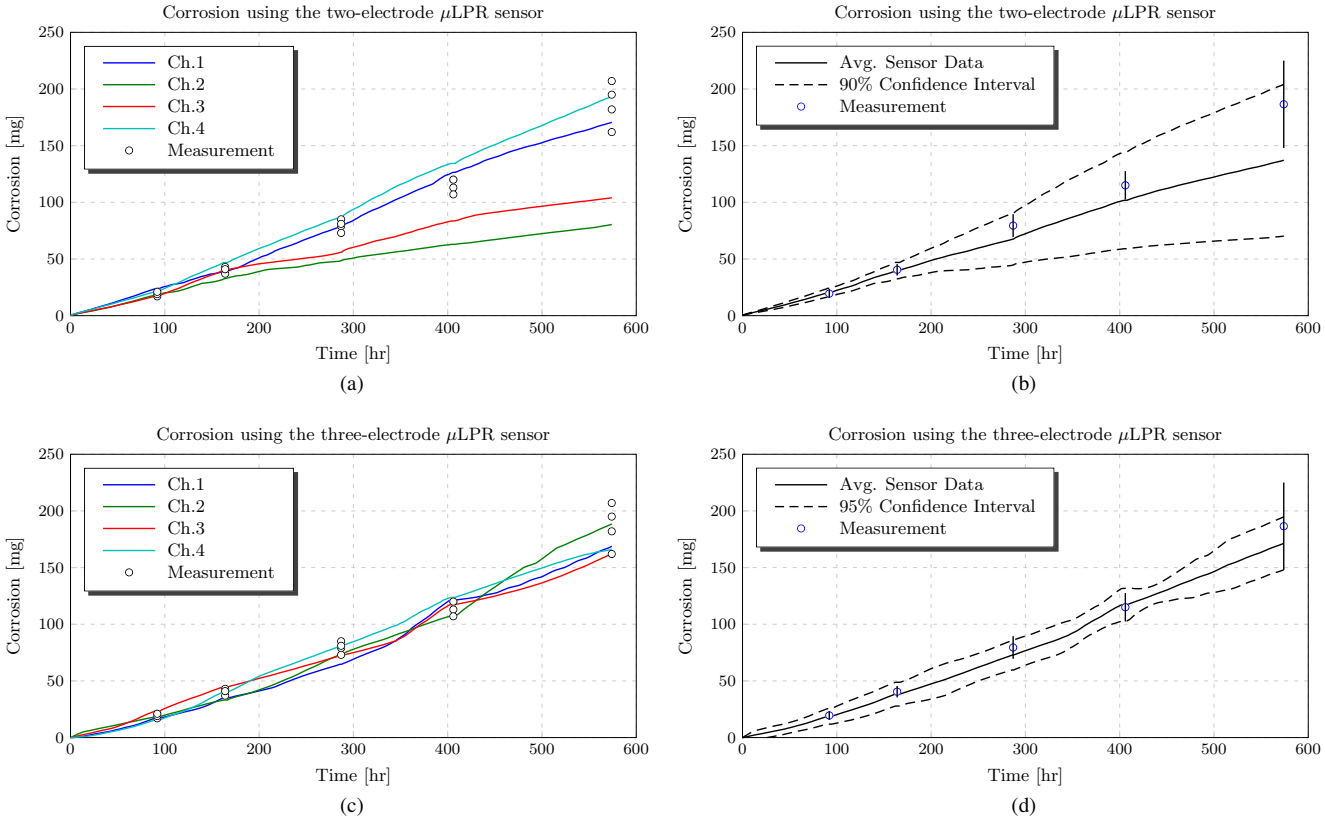


Figure 6. Corrosion vs. time for (a) four two-electrode μLPR sensor made from AA7075-T6, (b) the corresponding average with a 90% confidence interval, (c) corrosion vs. time for a three-electrode μLPR sensor made from nickel and (d) the corresponding average with a 95% confidence interval.

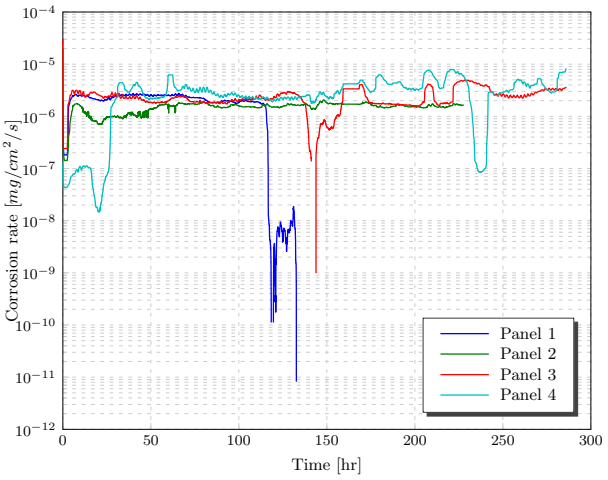


Figure 7. Computed corrosion rate vs. time.

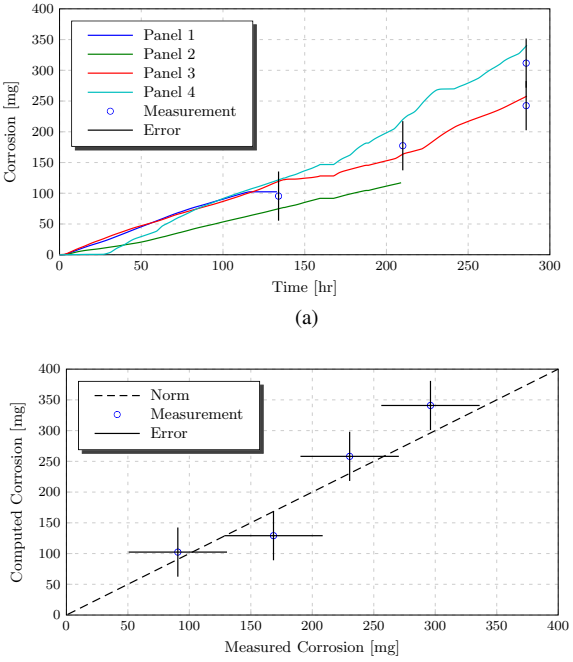


Figure 9. Plot of (a) computed corrosion vs. time and (b) measured vs. computed corrosion.

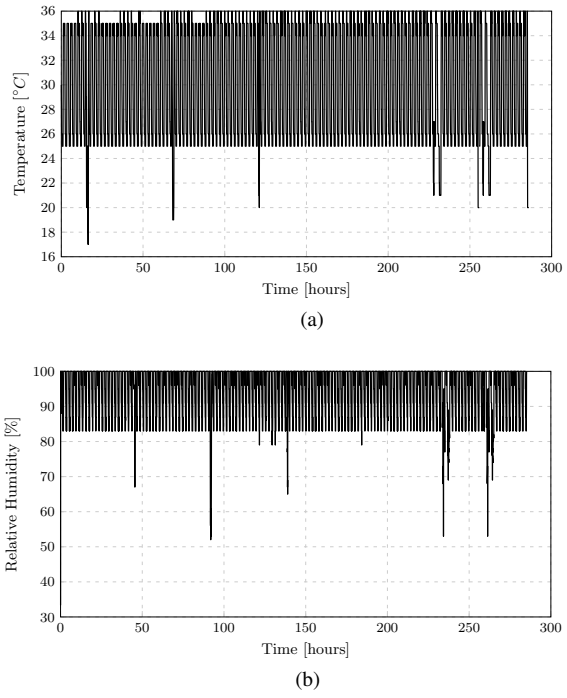


Figure 8. Plots of (a) temperature and (b) relative humidity vs. time.

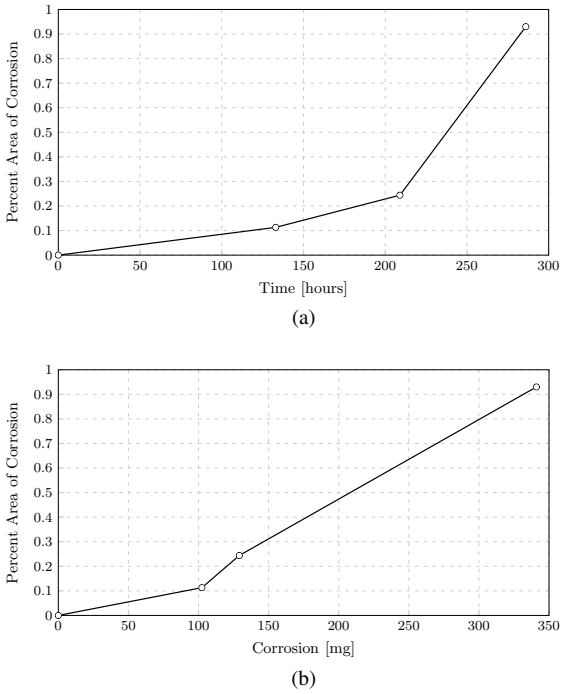


Figure 10. Percent area of corrosion vs. (a) time and (b) computed corrosion.

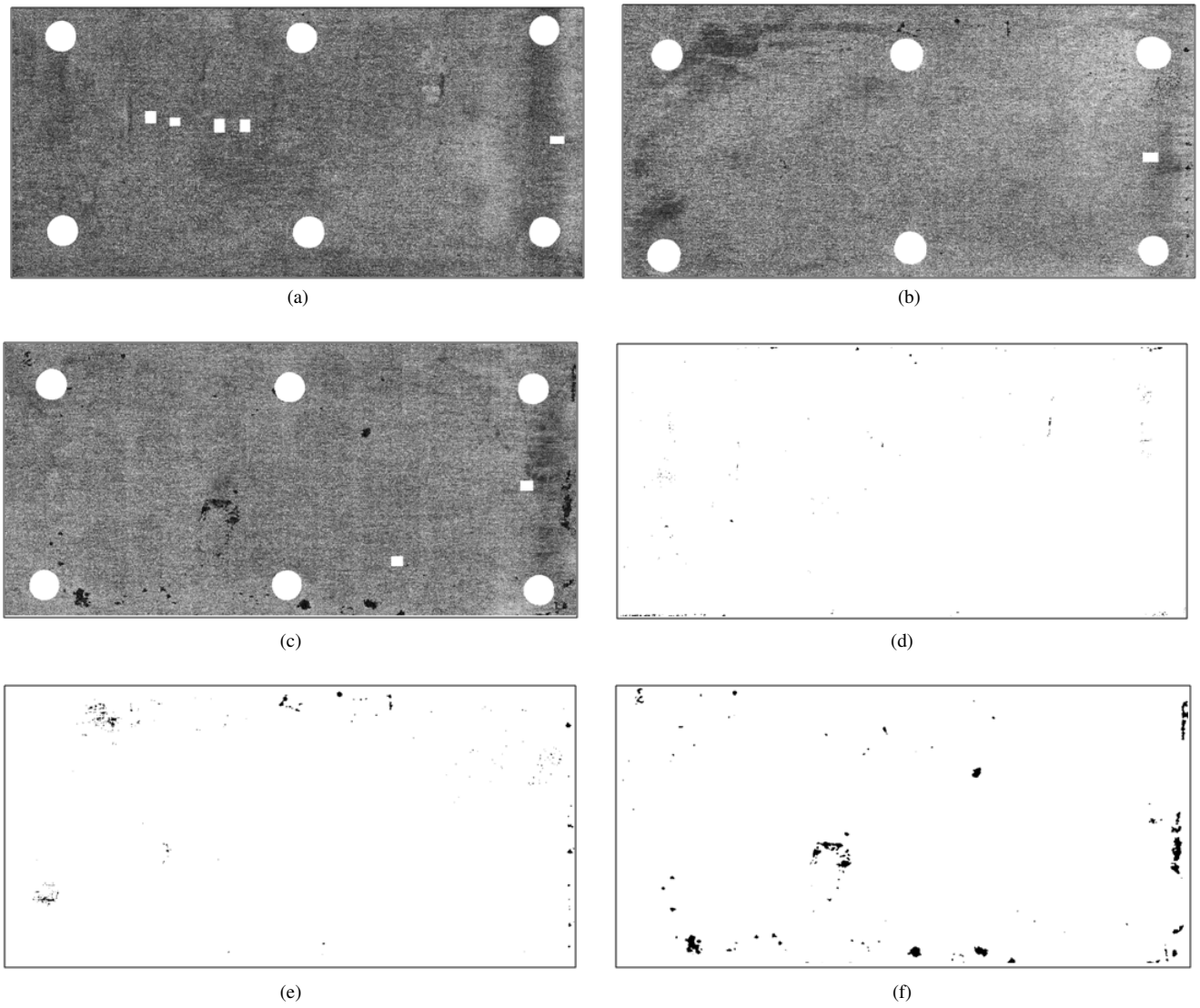


Figure 11. Original panel image with rivets and numbers removed for (a) 133 hours, (b) 209 hours, and (c) 286 hours of exposure time. Also shown is a binary image after filtering showing the percent area of corrosion for (d) 133 hours at 0.113%, (e) 209 hours at 0.244%, and (f) 286 hours at 0.93%.

ACKNOWLEDGMENT

All funding and development for the μLPR sensor and systems in the project has been part of the US government's SBIR programs. In particular: 1) Funding for the preparation of the initial system design and development was provided by the US Air Force under SBIR Phase II contract # F33615-01-C-5612 monitored by Dr. James Mazza; 2) Funding for the development and experimental set-up was provided by the US Navy under SBIR Phase II contract # N68335-06-C-0317 monitored by Dr. Paul Kulowitch; and 3) further improvements, scheduled field installations, and technology transition by the US Air Force under SBIR Phase II contract # FA8501-11-C-0012 and BAA/RIF contract # FA8650-12-C-0001 monitored by Mr. Feraidoon Zahiri.

NOMENCLATURE

β_a	V/dec	0.40	anodic Tafel constant
β_c	V/dec	0.15	cathodic Tafel constant
τ	s	-	time variable
$d\tau$	s	-	time step
k	-	-	sample index
t	s	-	time
t_0	s	-	initial time
z	-	3	electron loss
A_{corr}	cm^2	-	% area of corrosion
A_{image}	cm^2	-	% area of image
A_{rivets}	cm^2	-	% area of of rivets
A_{sen}	cm^2	4.233×10^{-2}	sensor area
AW	g/mol	2.899×10^{1}	atomic weight
$B^\star$	V/dec	4.95×10^{-2}	constant
B_{loss}	$\Omega \cdot g/cm^2/s$	1.170×10^{-4}	constant
E_a	V	-	applied potential
E_{corr}	V	-	corrosion potential
I_a	A/cm^2	-	applied current
I_{corr}	A/cm^2	-	corrosion current
F	C/mol	9.649×10^{4}	Faraday's constant
M_{loss}	g/cm^2	-	mass loss
N	-	-	total samples
P_{area}	-	-	Percent area of corrosion
R_{loss}	$g/cm^2/s$	-	corrosion rate
R_p	Ω	-	polarization resistance
T_s	s	60	sample period

REFERENCES

Bockris, J. O., Reddy, A. K. N., & Gambola-Aldeco, M. (2000). *Modern electrochemistry 2a. fundamentals of electrodics* (2nd ed.). New York: Kluwer Academic/Plenum Publishers.

Buchheit, R. G., Hinkebein, T., Maestas, L., & Montes, L. (1998, March 22-27). Corrosion monitoring of concrete-lined brine service pipelines using ac and dc electrochemical methods. In *Corrosion 98.* San Diego, Ca.

Burstein, G. T. (2005, December). A century of tafel's equation: 1905-2005. *Corrosion Science*, *47*(12), 2858-2870.

G102, A. S. (1994). Standard practice for calculation of corrosion rates and related information from electrochemical measurements. *Annual Book of ASTM Standards*, *03.02*.

G59, A. S. (1994). Standard practice for conducting potentiodynamic polarization resistance measurements. *Annual Book of ASTM Standards*, *03.02*.

Harris, S. J., Mishon, M., & Hebbron, M. (2006, October). Corrosion sensors to reduce aircraft maintenance. In *Rto avt-144 workshop on enhanced aircraft platform availability through advanced maintenance concepts and technologies.* Vilnius, Lithuania.

Herder, P., & Wijnia, Y. (2011). *Asset management: The state of the art in europe from a life cycle perspective* (T. van der Lei, Ed.). Springer.

Huston, D. (2010). *Structural sensing, health monitoring, and performance evaluation* (B. Jones & W. B. S. J. Jnr., Eds.). Taylor and Francis.

Introduction to corrosion monitoring. (2012, August 20). Online. Available from `http://www.alspi.com/introduction.htm`

Kossowsky, R. (1989). *Surface modification engineering* (Vol. 1). Boca Raton, Florida: CRC Press, Inc.

Twomey, M. (1997). Inspection techniques for detecting corrosion under insulation. *Material Evaluation*, *55*(2), 129-133.

Wagner, C., & Traud, W. (1938). *Elektrochem*, *44*, 391.

BIOGRAPHIES

Douglas W. Brown is the Senior Systems Engineer for Analatom, Inc. He received the bachelor of science degree in electrical engineering from the Rochester Institute of Technology and his master of science and doctor of philosophy degrees in electrical engineering from the Georgia Institute of Technology. Dr. Brown has ten years of experience developing and maturing Prognostics & Health Management (PHM) and fault-tolerant control systems in avionics application. He is a recipient of the National Defense Science and Engineering Graduate (NDSEG) Fellowship and has received several best-paper awards for his work in PHM and fault-tolerant control.

Richard J. Connolly is the Senior Research Engineer for Analatom, Inc. He completed his bachelor of science and doctor of philosophy degree in chemical and biomedical engineering at the University of South Florida. Dr. Connolly is

a fellow of the National Science Foundation and is regarded as an expert in interfacing of engineering devices with skin. He has extensive experience in bioelectrics, electrochemistry, and data analysis. Much of this experience was gained while performing bioelectric data collection on human and animal models. During his tenure at Analatom he has overseen testing and validation of the μLPR technology for aerospace and civil engineering applications.

Bernard Laskowski is the President and Senior Research Scientist at Analatom since 1981. He received the licentiaat and doctor of philosophy degrees in physics from the University of Brussels in 1969 and 1974, respectively. Dr. Laskowski has published over 30 papers in international refereed journals in the fields of micro-physics and microchemistry. As president of Analatom, Dr. Laskowski has managed 93 university, government, and private industry contracts, receiving a U.S. Small Business Administration Administrator's Award for Excellence.

Margaret Garvan received her master of science degree in electrical and computer engineering (ECE) from the Georgia Institute of Technology, and bachelor of science in electrical engineering from the University of Florida. She is currently a Ph.D. candidate and graduate research assistant at the Georgia Institute of Technology. Her research is focused on intelligent machine learning, and methodologies for diagnostics and prognostics for structural health monitoring.

Honglei Li received her master of science degree in electrical and computer engineering (ECE) from the Georgia Institute of Technology, and in Instrumental Engineering from Shanghai Jiao Tong University respectively. She is currently a graduate research assistant at Intelligent Control Systems Laboratory, working on her doctoral degree in ECE at the Georgia Institute of Technology. Her current research is focused on intelligent machine learning, methodologies for prognostics and structural health monitoring and health management, as well as asset life-cycle and risk management.

Vinod S. Agarwala Dr. Vinod S. Agarwala is a recently retired from the U.S. Civil Service as a Navy senior staff scientist and Esteemed Fellow of Naval Air Systems Command, Patuxent River, MD. He received a bachelor of science degree in Physics, Chemistry and Mathematics, two masters of science degrees, and a doctor of philosophy degree in Chemistry and Metallurgy from Banaras Hindu University (India) and Massachusetts Institute of Technology (USA). He has 35 years of distinguished civil service with major contributions in aircraft research and development technologies; he was awarded Department of The Navy Superior Civilian Service Medal. From 2006 - 2008, he was Associate Director at the U. S. Office of Naval Research Global - London, UK. There he served as an international agent for U.S. Navy with a mission to encourage international collaboration in Science and Technology through priority R&D in support of U.S. Naval forces.

George Vachtsevanos is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, a M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at the Georgia Institute of Technology where faculty and students are conducting research in intelligent control, neurotechnology and cardiotechnology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 240 technical papers and is a senior member of IEEE. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.

On the Use of Particle Flow to Enhance the Computational Performance of Particle-Filtering-based Prognostics

Javier A. Oliva[1], Torsten Bertram[2]

[1,2] *Institute of Control Theory and Systems Engineering, Technische Universität Dortmund, Germany*
javier.oliva@tu-dortmund.de
torsten.bertram@tu-dortmund.de

ABSTRACT

Prognostic approaches based on particle filtering employ physical models in order to estimate the remaining useful life (RUL) of systems. To this aim a set of particles is used to first estimate the degradation state of the system and then to predict the distribution of the RUL through simulation. The computational complexity of this approach is a function of the number of particles used in the state estimation and of the time each particle needs to simulate the RUL. It is therefore clear that enhancing the computational performance of this approach requires reducing the number of particles. In this paper we investigate the applicability and suitability of the particle flow particle filter for particle-filtering-based prognostics. The estimation of the remaining driving range (RDR) of an electric vehicle is used as the case study to illustrate the improvement in computational performance of the proposed approach in comparison to the standard particle filter.

1. INTRODUCTION

Model-based prognostic approaches have gained in importance during the last decade due to their versatility and ease of implementation in practical engineering applications. From the methodologies available in the literature, a model-based framework using particle filters (PF) has emerged as a solid solution for many prognostics applications. Particle-filtering based approaches for prognostics employ physics-based models in order to estimate the remaining useful life (RUL) of systems or components. To this aim a set of discrete weighted samples, known as particles, is used to first estimate the degradation state of the system or component and then to predict a distribution of the RUL by propagating the set of particles forward in time through simulation until an established failure threshold is reached. The computational complexity of this approach is a function of the number of particles used in the state estimation and of the time each particle needs to simulate the RUL. It is therefore clear that enhancing the computational performance of this approach requires minimizing the number of particles used without sacrificing the accuracy of both the estimation of the degradation state and the prediction of the RUL distribution. An approach that aims to solve this issue is introduced by (Daigle & Goebel, 2010). This approach is based on the Unscented Transform (UT) (Julier & Uhlmann, 2004), in which the particles are chosen deterministically instead of using a random sampling method. Although this method is more computationally efficient than standard particle filters, the UT may only be applied to nonlinear systems where all sources of noise are Gaussian; otherwise this approach should not be used. In this paper we investigate the use and the suitability of a well known variation of the particle filter based on particle flow and optimal transport methods. The main idea behind this approach is to reduce the number of particles needed in the particle filter by introducing a particle flow, in which the particles are progressively transported without needing to randomly sample from any distribution. This allows us to optimally move the particles to the correct locations according to the Bayes' rule, reducing in this way the number of particles needed and thereby the computational effort in both the estimation and the prediction step. To the best of our knowledge the present study is the first in applying the the particle flow particle filter in model-based prognostics. This paper evaluates the use of the particle flow, which until now has been just investigated in filtering problems of nonlinear systems (Daum & Huang, 2008), with the aim of presenting a computationally efficient alternative to state of the art simulation-based approaches, namely UKF (Daigle & Goebel, 2010) and PF (Orchard & Vachtsevanos, 2010) based approaches, for reducing the number of simulations and therefore the simulation time in the prediction step of model-based prognostics. We use the remaining driving range (RDR) estimation of an electric vehicle (Oliva, Weihrauch, & Bertram, 2013) as the case of study for illustrating and validating the enhancement in the computational performance of the presented approach in comparison to the standard particle filter.

Javier A. Oliva et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The remainder of this paper is organized as follows. Section 2 formulates the RUL estimation problem in the context of particle filters. Section 3 explains in detail the theoretical foundations of the particle flow particle filter (PFPF) and afterwards presents the steps needed for its implementation within the prognostics framework presented in section 2. In section 4 the case of study used for validating the proposed approach is described. Section 5 presents the experimental and simulation results. Finally, section 6 concludes the findings of this work and provides an outlook on our future work.

2. PARTICLE-FILTERING BASED RUL ESTIMATION

This section is concerned with formulating the RUL estimation problem and briefly explains the particle-filtering-based framework for prognostics employed in this work.

2.1. Problem Statement

Consider the following nonlinear system represented, in a discrete-time form by

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k\right) \\ \mathbf{y}_k &= \mathbf{h}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k, \mathbf{w}_k\right), \end{aligned} \tag{1}$$

where $\mathbf{x}_k$ is the state vector, $\mathbf{w}_k$ is the parameter vector, $\mathbf{v}_k$ is the process noise vector, $\mathbf{u}_k$ is the input vector, $\mathbf{y}_k$ is the output vector and $\mathbf{n}_k$ is the measurement noise vector. The terms $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ stand for the state and output function, respectively. The system exhibits a degradation which accumulates in time until a deterministic degradation threshold $\boldsymbol{T}(\mathbf{x})$ is reached, at which the system fails. The degradation of the system is attributed to the environment and to the operation conditions. The RUL estimation problem is concerned with first estimating the degradation state of the system and then to predict its future operation conditions in order to determine the distribution of the time at which the performance of the system fails to fulfill its tasks, i.e. the time at which the threshold is exceeded. Thus, $\boldsymbol{T}(\mathbf{x}) = 1$ if the system fails and $\boldsymbol{T}(\mathbf{x}) = 0$, otherwise. The RUL is a random variable that is influenced by many sources of uncertainty. The lack of knowledge about the state variables, the noise presented in the measurements or the randomness of the operation environment, are some of the factors that largely contribute to the uncertainty of the RUL. Therefore, properly predicting the RUL requires accounting for these sources of uncertainty. In the context of particle filters the RUL estimation proceeds basically in two phases, namely the *state estimation* (I) and the RUL *prediction* (II), as shown in Fig. 1. For the sake of clarity, Fig. 1 depicts the RUL estimation of just one particle.

In the first phase the PF recursively approximates the posterior probability $p(\mathbf{x}_k|\mathbf{Y}_k)$ of the state variables by a set of N_x weighted particles $\mathbf{S}_k = \left\{\mathbf{x}_k^i, w_k^i\right\}_{i=1}^{N_x}$. Here $\mathbf{x}_k^i$ is the set of particles representing the state space, w_k^i are the associated importance weights and $\mathbf{Y}_k = \mathbf{y}_{0:k}$ is the set of all measurements done until time k. Each particle is sampled from an *a priori* estimation of the state space and it is propagated through the function $\mathbf{f}(\cdot)$ in the *prediction step*. Then, the value of each particle is updated from measurements through the output function $\mathbf{h}(\cdot)$ in the *measurement update step*. In this step the weight of each particle is updated according to the likelihood of a new measurement given the particle. Afterwards the *resampling step* occurs. The idea behind this step is to duplicate those particles with large weights and to eliminate those with small weights.

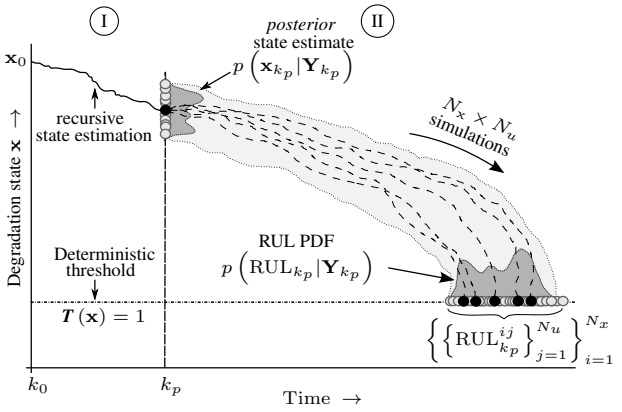


Figure 1. Particle-filtering based RUL estimation approach.

In this way the so called *particle degeneracy* (Daum & Huang, 2011) can be overcome. Particle degeneracy, i.e the situation in which all but few particles have negligible weights leads to a poor approximation of the state variables and, since most weights are close to zero, valuable computational effort is wasted by updating insignificant particles. Finally, the probability distribution of the state variables at time k is approximated by

$$p(\mathbf{x}_k|\mathbf{Y}_k) \approx \frac{1}{N_x}\sum_{i=1}^{N_x} w_k^i \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right) \tag{2}$$

where $\delta(\cdot)$ describes the Dirac delta function located at $\mathbf{x}_k^i$. The posterior state estimate establishes the starting point for the second phase, in which the particle filter is employed for predicting the RUL at given time k_p. To this aim the posterior estimate $p(\mathbf{x}_{k_p}|\mathbf{y}_{k_p})$ is set as initial condition.

By assuming that the set of particles $\mathbf{S}_k$ accurately represents the unknown states at the time of prediction, it is possible to approximate the probability density function of system states at any time $k_p + m$ in the future by means of the law of total probabilities (Orchard & Vachtsevanos, 2010)

$$\hat{p}\left(\mathbf{x}_{k_p+m}|\hat{\mathbf{x}}_{k_p:k_p+m-1}\right) \approx \sum_{i=1}^{N_x} w_{k_p+m-1}^i \hat{p}\left(\hat{\mathbf{x}}_{k_p+m}^i|\hat{\mathbf{x}}_{k_p+m-1}^i\right). \tag{3}$$

To account for the fact, that during the prediction the shape of the states probability distribution may change, due to noise and process nonlinearities, Eq.(3) requires the set of weights to be updated at each iteration. However, during the prediction step no new measurements, which could serve for updating the weights, can be acquired. This implies that an update procedure for the particle weights, as it would happen in a typical filtering problem, cannot be carried out. This issue is addressed by assuming the weights as invariant from the time k_p to $k_p + m$. This assumption is justified by considering the uncertainty added by model inaccuracies or by the ignorance about future operation conditions to be large in comparison to the uncertainty which comes from considering constant particle weights. In this way, the set of weighted particles $\mathbf{S}_{k_p}$ is simply propagated forward into the future by simulating the behavior of the system as reaction to a future operation condition, until the determined failure condition is reached.

Once all particles have reached this point, i.e. $\boldsymbol{T}^i_{k_p} = 1$, the $\mathrm{RUL}^i_{k_p}$ of each particle is determined and combined with its weight $w^i_{k_p}$ to approximate $p\left(\mathrm{RUL}_{k_p}|\mathbf{Y}_{k_p}\right)$ as follows

$$p\left(\mathrm{RUL}_{k_p}|\mathbf{Y}_{k_p}\right) \approx \sum_{i=1}^{N_x} w^i_{k_p} \mathrm{RUL}^i_{k_p}. \tag{4}$$

The RUL prediction, as formulated in Eq.(4), requires propagating the set of particles through a single hypothesized predicted profile of the future operation conditions of the system. However, such a propagation accounts just for the uncertainty introduced in the state estimation step but it does not consider the uncertainty related to the predicted operation profile. Taking this uncertainty into account would require propagating the set of particles through multiple predicted profiles, and not through a single one. Thus, the computational complexity of such a prediction becomes a function of $N_x \times N_u$ (Daigle, Saxena, & Goebel, 2012), where N_u is the number of predicted operation profiles. The set of weighted particles is then propagated through multiple profiles until all particles along all predicted profiles, have reached the threshold, i.e. $\boldsymbol{T}^{ij}_{k_p} = 1$. Here j represents each predicted operation profile. Accordingly, the probability distribution $p\left(\mathrm{RUL}_{k_p}|\mathbf{Y}_{k_p}\right)$ is approximated by

$$p\left(\mathrm{RUL}_{k_p}|\mathbf{Y}_{k_p}\right) \approx \frac{1}{N_u} \sum_{j=1}^{N_u} \sum_{i=1}^{N_x} w^i_{k_p} \mathrm{RUL}^{ij}_{k_p}. \tag{5}$$

It must be noted that all predicted profiles are equally weighted by means of $\frac{1}{N_u}$.

3. PARTICLE FLOW PARTICLE FILTER

From the previous section it can be inferred that the computational performance of the particle-filter-based RUL estimation approach can be enhanced through the reduction of the particles employed during the estimation step and therefore during the prediction step. However, this cannot be done straightforward specially in those systems where the dimensionality of the state space is high. This problem becomes more significant in a joint state/parameter estimation since the dimensionality of the state space can increase considerably. In this paper we aim to investigate the suitability of an approach for reducing the number of particles needed in the estimation of the state space without sacrificing the accuracy of the state estimation.

Standard particle filters might reduce the computational performance of the prognostics algorithm during the estimation step by wasting computational resources during the propagation of those particles with negligible weights. Furthermore, since either particles with very low weight or duplicated particles have to be propagated forward in time until they reach the predefined threshold, additional resources might be wasted during the prediction step of the prognostics framework.

The approach presented in this paper aims to overcome the aforementioned issues by implementing an update schema, which progressively transforms the prior $p\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right)$ into the posterior state estimate $p\left(\mathbf{x}_k|\mathbf{Y}_k\right)$ by smoothly moving the particles in an optimal manner as new measurements become available without needing to employ any resampling algorithm. This is achieved by solving a differential equation to determine the flow of particles in the state space as they migrate from the prior to the posterior distribution. In a generic Bayesian framework, the posterior $p\left(\mathbf{x}_k|\mathbf{Y}_k\right)$ is obtained in the prediction step by a single computation of the Bayes' rule given by

$$\overbrace{p\left(\mathbf{x}_k|\mathbf{Y}_k\right)}^{\text{posterior}} = \frac{\overbrace{p\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right)}^{\text{prior}} \overbrace{p\left(\mathbf{y}_k|\mathbf{x}_k\right)}^{\text{likelihood}}}{\underbrace{\int_{\mathbb{R}} p\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right) p\left(\mathbf{y}_k|\mathbf{x}_k\right) \mathrm{d}x_k}_{\text{normalization factor}}}. \tag{6}$$

By denoting a new set of density functions given by

$$\begin{aligned} \psi\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_k\right) &= p\left(\mathbf{x}_k|\mathbf{Y}_k\right) \\ g\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_{k-1}\right) &= p\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right) \end{aligned} \tag{7}$$

it is possible to compute $\psi\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_k\right)$ in a *B*-fold recursive manner by progressively introducing the likelihood density, here denoted as $l\left(\mathbf{y}_k|\mathbf{x}_k\right)$, such that the prior $g\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_{k-1}\right)$ gradually deforms into $g\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_{k-1}\right) l\left(\mathbf{y}_k|\mathbf{x}_k\right)$. This can be achieved by using a homotopy of the form

$$\overbrace{\psi\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_k\right)}^{\text{posterior}} = \frac{\overbrace{g\left(\mathbf{x}_{k,\lambda}|\mathbf{Y}_{k-1}\right)}^{\text{prior}} \overbrace{l\left(\mathbf{y}_k|\mathbf{x}_{k,\lambda}\right)^{\lambda}}^{\text{likelihood}}}{\underbrace{K_{k,\lambda}}_{\text{normalization factor}}}, \tag{8}$$

where $\lambda \in [0,1]$ is the progression parameter and the term $l\left(\mathbf{y}_k|\mathbf{x}_{k,\lambda}\right)^{\lambda}$ is understood as an incremental likelihood. Thus, Eq. (8) represents the prior when $\lambda = 0$ and the posterior when $\lambda = 1$. The number of iterations in the recursion, namely B, depends on the step size $\Delta\lambda$, which determines the rate at which $\lambda_{0\to1}$. The way $l\left(\mathbf{y}_k|\mathbf{x}_{k,\lambda}\right)^{\lambda}$ is incrementally incorporated into the Bayes' update step can be seen in the Algorithm 1. For the sake of clarity, from now on we express the states variables as $\mathbf{x}_\lambda$ instead as $\mathbf{x}_{k,\lambda}$. This is due to the fact that the evolution of the probability distribution as $\lambda_{0\to1}$ always occurs at the discrete time step k. In order to avoid numerical issues the log-density of Eq. (8) is applied yielding to

$$\Psi\left(\mathbf{x}_\lambda\right) = G\left(\mathbf{x}_\lambda\right) + \lambda L\left(\mathbf{x}_\lambda\right) - \log K_\lambda, \tag{9}$$

where the posterior is given by $\Psi\left(\mathbf{x}_\lambda\right) = \log\psi\left(\mathbf{x}_\lambda|\mathbf{Y}_k\right)$, the prior is represented by $G\left(\mathbf{x}_\lambda\right) = \log g\left(\mathbf{x}_\lambda|\mathbf{Y}_{k-1}\right)$ and the likelihood is $L\left(\mathbf{x}_\lambda\right) = \log l\left(\mathbf{y}_k|\mathbf{x}_\lambda\right)$. The evolution of the probability distribution given by Eq. (9) in the pseudo-time is known as *log-homotopy* (Daum & Huang, 2008). As it can be seen in Fig. 2, the task of this homotopy is to move the particles through a sequence of densities from the prior to the posterior as λ continuously increases from zero to one.

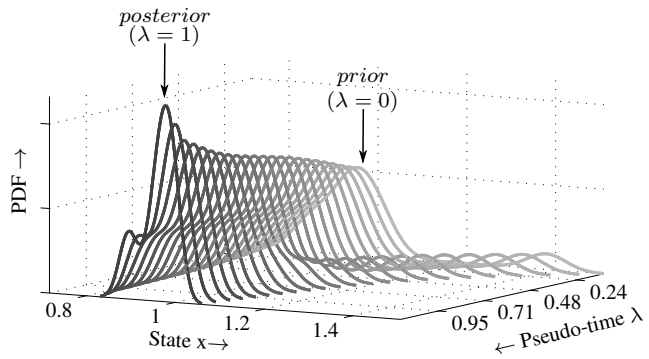


Figure 2. Evolution of the probability distribution from the prior at $\lambda = 0$ to the posterior at $\lambda = 1$.

As it can be observed in Fig. 3, it becomes necessary to find a flow $\frac{\mathrm{dx}}{\mathrm{d}\lambda}$ that dictates the motion of particles as they move following the log-homotopy given by Eq. (9).

To this aim we differentiate Eq. (9) with respect to λ

$$\frac{\partial\Psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} = L\left(\mathbf{x}_\lambda\right) - \frac{\mathrm{d}}{\mathrm{d}\lambda}\log K_\lambda. \tag{10}$$

Replacing the left hand side of Eq. (10) by the logarithm identity

$$\frac{\partial\Psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} = \frac{1}{\psi\left(\mathbf{x}_\lambda\right)}\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} \tag{11}$$

and multiplying both sides by $\psi\left(\mathbf{x}_\lambda\right)$ yields to

$$\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} = \psi\left(\mathbf{x}_\lambda\right)\left[L\left(\mathbf{x}_\lambda\right) - \frac{\mathrm{d}\log K_\lambda}{\mathrm{d}\lambda}\right]. \tag{12}$$

A way to find the desired flow $\frac{\mathrm{dx}}{\mathrm{d}\lambda}$ is by considering that the particles move, as $\lambda_{0\to1}$, obeying the following stochastic differential equation (SDE)

$$\mathrm{d}\mathbf{x}_\lambda = \boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)\mathrm{d}\lambda + \boldsymbol{\eta}\left(\mathbf{x}_\lambda\right)\mathrm{d}\boldsymbol{\xi}_\lambda, \tag{13}$$

where $\mathbf{x}_\lambda$ is the particle position at given time k and pseudo-time λ, $\boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)$ can be understood as a *vector field* that induces the motion of particles from the prior to the posterior distribution, $\eta\left(\cdot\right)$ is a multiplicative noise matrix and $\boldsymbol{\xi}_\lambda$ is a noise resulting from the randomness of process.

By considering $\frac{\mathrm{dx}}{\mathrm{d}\lambda}$ to be given by $\boldsymbol{\zeta}\left(x_\lambda\right)$, the desired particle flow can be obtained by using the conditional probability density $\psi\left(\mathbf{x}_\lambda\right)$ together with the forward Kolmogorov equation, also known as the Fokker-Planck-Kolmogorov (FPK) equation. In this context the FPK equation is employed to relate the flow $\frac{\mathrm{dx}}{\mathrm{d}\lambda}$ of a particle with the evolution of $\psi\left(\mathbf{x}_\lambda\right)$ as $\lambda_{0\to1}$ under the influence of drift and diffusion processes.

The FPK equation can be written as

$$\begin{aligned}\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} &= \overbrace{-\mathrm{tr}\left[\frac{\partial}{\partial\mathbf{x}_\lambda}\left(\boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)\psi\left(\mathbf{x}_\lambda\right)\right)\right]}^{\text{drift}} + \\ &\quad + \overbrace{\frac{1}{2}\mathrm{tr}\left[\frac{\partial}{\partial\mathbf{x}_\lambda}\left(\mathbf{Q}\left(\mathbf{x}_\lambda\right)\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda}\right)\right]}^{\text{diffusion}},\end{aligned} \tag{14}$$

where $\mathbf{Q}\left(\mathbf{x}_\lambda\right) = \boldsymbol{\eta}\left(\mathbf{x}_\lambda\right)\boldsymbol{\eta}^{\mathrm{T}}\left(\mathbf{x}_\lambda\right)$ is the process covariance matrix and $\mathrm{tr}\left(\cdot\right)$ states for the *trace* of $\left(\cdot\right)$.

Reformulating Eq. (14) in a more proper way yields

$$\begin{aligned}\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\lambda} &= -\mathrm{tr}\left[\psi\left(\mathbf{x}_\lambda\right)\frac{\partial\boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)}{\partial\mathrm{x}_\lambda} + \boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)^{\mathrm{T}}\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda}\right] + \\ &\quad + \frac{1}{2}\mathrm{div}\left(\mathbf{Q}\left(\mathbf{x}_\lambda\right)\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda}\right) \\ &= -\boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)^{\mathrm{T}}\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda} - \psi\left(\mathbf{x}_\lambda\right)\mathrm{tr}\left(\frac{\partial\boldsymbol{\zeta}\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda}\right) + \\ &\quad + \frac{1}{2}\mathrm{div}\left(\mathbf{Q}\left(\mathbf{x}_\lambda\right)\frac{\partial\psi\left(\mathbf{x}_\lambda\right)}{\partial\mathbf{x}_\lambda}\right),\end{aligned} \tag{15}$$

where $\mathrm{div}\left(\cdot\right)$ states for the *divergence* of $\left(\cdot\right)$. As it can be seen, Eq. (12) and Eq. (15) are equivalent. Thus, equating them and by dividing both sides by $\psi\left(\mathbf{x}_\lambda\right)$ we can write

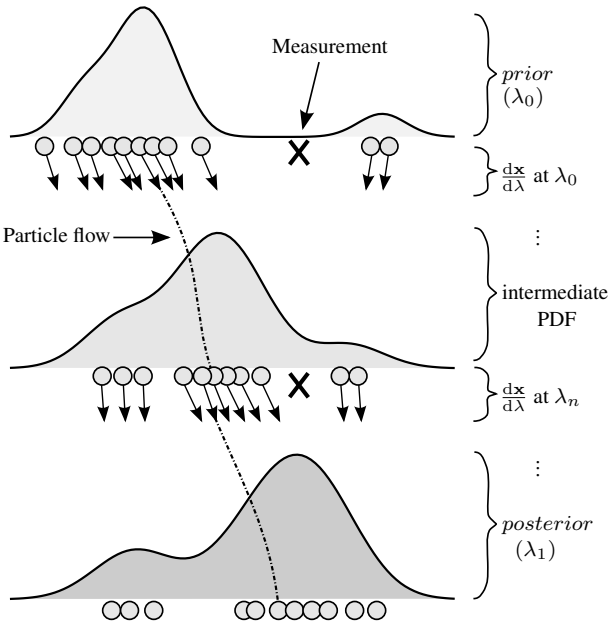


Figure 3. Particle flow at different values of λ.

$$L(\mathbf{x}_\lambda) - \frac{\mathrm{d}\log K_\lambda}{\mathrm{d}\lambda} = -\boldsymbol{\zeta}(\mathbf{x}_\lambda)^{\mathrm{T}} \frac{1}{\psi(\mathbf{x}_\lambda)} \frac{\partial \psi(\mathbf{x}_\lambda)}{\partial \mathrm{x}_\lambda} + -\mathrm{tr}\left(\frac{\partial \boldsymbol{\zeta}(\mathbf{x}_\lambda)}{\partial \mathrm{x}_\lambda}\right) + + \frac{1}{2\psi(\mathbf{x}_\lambda)} \mathrm{div}\left(\mathbf{Q}(\mathbf{x}_\lambda) \frac{\partial \psi(\mathbf{x}_\lambda)}{\partial \mathrm{x}_\lambda}\right), \tag{16}$$

under the assumption that $\psi(\mathbf{x}_\lambda)$ is nowhere vanishing. The desired particle flow is found by solving Eq. (16) wrt. $\boldsymbol{\zeta}(\mathbf{x}_\lambda)$.

To this aim we first compute the gradient wrt. $\mathbf{x}_\lambda$. This yields to a system of partially differential equations (PDEs) with the same number of unknowns and equations given by

$$\frac{\partial L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda} = -\boldsymbol{\zeta}^{\mathrm{T}}(\mathbf{x}_\lambda) \frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} - \frac{\partial \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda} \frac{\partial \boldsymbol{\zeta}(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda} + -\frac{\partial}{\partial \mathbf{x}_\lambda}\left[\mathrm{tr}\left(\frac{\partial \boldsymbol{\zeta}(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda}\right)\right] + + \frac{\partial}{\partial \mathbf{x}_\lambda}\left[\frac{1}{2\psi(\mathbf{x}_\lambda)} \mathrm{div}\left(\mathbf{Q}(\mathbf{x}_\lambda) \frac{\partial \psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda}\right)\right]. \tag{17}$$

There are many methods to solve the system of PDE's given by Eq. (17) (Daum & Huang, 2010). In this work we employ the approach presented by (Daum & Huang, 2013) in which it is assumed that both the process noise matrix $\mathbf{Q}(\mathbf{x}_\lambda)$ and the vector field given by $\boldsymbol{\zeta}(\mathbf{x}_\lambda)$ are chosen such that sum of the last three terms of Eq. (17) is zero. In this manner the system of PDE's is drastically simplified yielding to the following equation

$$\frac{\partial L(\mathbf{x}_\lambda)}{\partial \mathbf{x}} = -\boldsymbol{\zeta}^{\mathrm{T}}(\mathbf{x}_\lambda) \frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} \tag{18}$$

As stated by (Daum & Huang, 2013), if it is assumed that $\frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}$ is non-singular, the solution of Eq. (18) for $\boldsymbol{\zeta}(\mathbf{x}_\lambda)$ can be computed as

$$\boldsymbol{\zeta}(\mathbf{x}_\lambda) = -\left[\frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}\right]^{-1} \left[\frac{\partial L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda}\right]^{\mathrm{T}}. \tag{19}$$

The task now is to compute the terms of the right hand side of Eq. (19). First, the Hessian $\frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}$ can be obtained in closed form by differentiating twice Eq. (9) wrt. $\mathbf{x}_\lambda$

$$\frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} = \frac{\partial^2 G(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} + \lambda \frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}. \tag{20}$$

In this work we use a hybrid approach for computing Eq. (20) in which the Hessian $\frac{\partial^2 G(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}$ is approximated by

$$\frac{\partial^2 G(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} \approx -\hat{\mathbf{S}}_{N_x}^{-1}, \tag{21}$$

where $\hat{\mathbf{S}}_{N_x}$ is the sample covariance matrix (SCM) of the prior distribution computed from the set of N_x particles. The SCM offers an unbiased estimate of the true covariance matrix. However, it has to be noted that if the number of particles employed is smaller than the number of states to be estimated the SCM may suffer from high variance. To overcome this issue the Kronecker product expansion can be used to estimate the covariance matrix in high dimensional spaces (Tsiligkaridis & Hero, 2013).

If it is assumed that the prior $g(\cdot)$ is represented by a Gaussian distribution, then the approximation given by Eq. (21) is exact. For practical purposes the likelihood function $l(\cdot)$ can be assumed to follow an univariate or a multivariate Gaussian distribution depending on the dimension of the output vector. Accordingly, $L(\mathbf{x}_\lambda)$ is expressed as

$$L(\mathbf{x}_\lambda) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{R}| - \frac{1}{2}\mathbf{z}_{k,\lambda}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{z}_{k,\lambda}, \tag{22}$$

where $\mathbf{z}_{k,\lambda} = (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_\lambda))$ and $\mathbf{R}$ is the covariance matrix of the measurement noise. Computing the gradient of Eq. (22) wrt. $\mathbf{x}_\lambda$ gives

$$\frac{\partial L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda} = \left[\frac{\partial \mathbf{h}(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda}\right]^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{z}_{k,\lambda} = \hat{\mathbf{H}}(\mathbf{x}_\lambda)^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_\lambda)), \tag{23}$$

where $\hat{\mathbf{H}}(\mathbf{x}_\lambda)$ is the linearized output matrix around $\mathbf{x}_\lambda$. Com-

puting the Hessian $\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}$ might be computationally expensive. We instead approximate it by computing the expected Hessian by means of the Monte Carlo approximation method as follows

$$\begin{aligned}\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2} &\approx E\left[\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}\right] \\ &\approx -\frac{1}{N_x}\sum_{i=1}^{N_x}\left[\frac{\partial \mathbf{z}_{k,\lambda}(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^{\mathrm{i}}}\right]^{\mathrm{T}}\mathbf{R}^{-1}\frac{\partial \mathbf{z}_{k,\lambda}(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^{\mathrm{i}}},\end{aligned} \tag{24}$$

where $E[\cdot]$ is the expected value with respect to the likelihood function. After having computed $\frac{\partial L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda}$ and $\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}$ both Eq. (19) and Eq. (20) can be evaluated in order to obtain the particle flow. As it can be seen, evaluating Eq. (20) requires computing the inverse of $\hat{\mathbf{S}}_{N_x}$, which can lead to numerical problems if $\hat{\mathbf{S}}_{N_x}$ is close to be singular. To overcome this issue we apply the matrix inversion lemma known as Woodbury's formula in order to invert Eq. (20) as follows

$$\begin{aligned}\left[\frac{\partial^2 \Psi(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}\right]^{-1} &= -\hat{\mathbf{S}}_{N_x} + \\ -\hat{\mathbf{S}}_{N_x}\lambda\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}&\left(I - \hat{\mathbf{S}}_{N_x}\lambda\frac{\partial^2 L(\mathbf{x}_\lambda)}{\partial \mathbf{x}_\lambda^2}\right)^{-1}\hat{\mathbf{S}}_{N_x}.\end{aligned} \tag{25}$$

Algorithm 1 summarizes the steps needed for implementing the presented particle flow particle filter for state estimation. It is worth noting that the rate at which $\lambda_{0\to 1}$ is determined by the step size $\Delta\lambda$. Numerical experiments presented by (Daum & Huang, 2013) have shown that employing a fixed step size, such as in the case of the Euler method, works properly just if the number of particles is high. Therefore, to reduce the number of particles employed a variable $\Delta\lambda$ has to be used. A proper strategy is to use a very small value of $\Delta\lambda$ at the beginning and to gradually increase it as $\lambda \to 1$, which makes sense, since the uncertainty a the beginning of the measurement update step is higher. We therefore use an exponentially increasing step size (George & Powell, 2006) given by

$$\Delta\lambda = 1 - \frac{1}{n^b}, \tag{26}$$

where n is the number of iteration and $b \in \left(\frac{1}{2}, 1\right]$. In the case of initial transient conditions the a small value of b can lead to a slower learning rate of the step size. The value of b should be chosen according to the desired rate of convergence of the step size.

4. CASE STUDY

For validating the applicability of the particle flow particle filter for prognostics we chose the remaining driving range (RDR) estimation of an electric vehicle (Oliva et al., 2013). In this context, the RDR estimation is concerned with predicting the power demand of the electric vehicle and identifying the distance that it can drive with the energy stored in its battery before recharging is required. To this aim we consider the battery state of charge (SOC) to be the indicator that determines the threshold condition.

Algorithm 1 Particle flow particle filter for state estimation

Initialization
Draw a set of particles $\left\{\mathbf{x}_0^i\right\}_{i=1}^{N_x}$ from the prior $p(\mathbf{x}_0)$
for $k = 1$ **to** ∞ **do**
  State prediction
  Propagate the particles through the system equation:
  $\mathbf{x}_{k|k-1}^i = \mathbf{f}\left(\mathbf{x}_{k-1}^i, \mathbf{u}_k, \mathbf{v}_k n; \mathbf{w}_k\right)$
  Initialize the pseudo-time $\lambda = 0$
  Set $\left\{\mathbf{x}_{k,\lambda}^i\right\}_{i=1}^{N_x} = \left\{\mathbf{x}_{k|k-1}^i\right\}_{i=1}^{N_x}$
  Measurement update:
  Propagate the particles through the output equation:
  $\mathbf{y}_{k|k}^i = \mathbf{h}\left(\mathbf{x}_k^i, \mathbf{u}_k, \mathbf{n}_k, \mathbf{w}_k\right)$
  while $\lambda \leq 1$ **do**
    Compute $\hat{\mathbf{S}}_{N_x}$ from $\left\{\mathbf{x}_{k,\lambda}^i\right\}_{i=1}^{N_x}$
    Calculate the state estimation from $\left\{\mathbf{x}_{k,\lambda}^i\right\}_{i=1}^{N_x}$
    $\hat{\mathbf{x}}_{k,\lambda} = \frac{1}{N_x}\sum_{i=1}^{N_x}\mathbf{x}_{k,\lambda}^i$
    Linearize $\mathbf{h}(\cdot)$ around $\hat{\mathbf{x}}_{k,\lambda}$ to compute $\hat{\mathbf{H}}$
    for $i = 1$ **to** N_x **do**
      Compute the flow $\boldsymbol{\zeta}\left(\mathbf{x}_{k,\lambda}^i\right)$ for each particle
      Set $\frac{\mathrm{d}\mathbf{x}_{k,\lambda}^i}{\mathrm{d}\lambda} = \boldsymbol{\zeta}\left(\mathbf{x}_{k,\lambda}^i\right)$
      Move the particles according their respective flow:
      $\mathbf{x}_{k,\lambda}^i = \mathbf{x}_{k,\lambda}^i + \Delta\lambda\frac{\mathrm{d}\mathbf{x}_{k,\lambda}^i}{\mathrm{d}\lambda}$
    end for
    Increment the pseudo-time $\lambda \leftarrow \lambda + \Delta\lambda$
  end while
  Update the state estimation:
  $\hat{\mathbf{x}}_k = \frac{1}{N_x}\sum_{i=1}^{N_x}\mathbf{x}_{k,\lambda}^i$
end for

Accordingly, the threshold is expressed as $\boldsymbol{T}(\mathrm{SOC})$. Thus, $\boldsymbol{T}(\mathrm{SOC}) = 1$ if $\mathrm{SOC}_{\min}$ (the minimum allowable state of charge) is reached and $\boldsymbol{T}(\mathrm{SOC}) = 0$, otherwise. The $\mathrm{SOC}_{\min}$ is usually dictated by the battery management system (BMS) of the electric vehicle in order to protect the battery cells from a possible total charge depletion.

4.1. Battery Model

We employ the model of a Li-ion cell shown in Fig. 4. The model combines the Kinetic Battery Model (Manwell & McGowan, 1994) for capturing the nonlinear effects in the battery capacity, such as the recovery and the rate capacity effect, with a second order equivalent circuit based model which

captures the dynamic response of the Li-ion cell. Furthermore, the combined model demands low computational effort, which makes it suitable for real-time applications. Even though the KiBaM was initially developed for lead acid batteries, it has been shown to be suitable for modeling the capacity behavior of Li-ion cells (Jongerden & Haverkort, 2009).

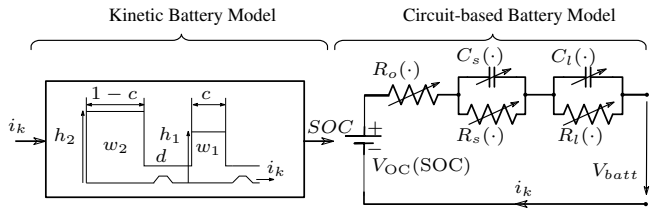


Figure 4. Combined battery model.

The Kinetic Battery Model abstracts the chemical processes of the battery discharge to its kinetic properties. The model assumes that the total charge of the battery is distributed with a capacity ratio $0 < c < 1$ between two charge wells. The first well contains the available charge and delivers it directly to the load. The second well supplies charge only to the first well by means of the parameter d. The rate of charge that flows from the second to the first well depends on both d and on the height difference between the wells $(h_2 - h_1)$. If the first well is empty, then the battery is considered to be fully discharged. By applying load to the battery, the charge in the first well is reduced, which leads to an increment in the height difference between both wells. After removing the load, certain amount of charge flows from the second well to the first well until the height of both wells is the same. In this way the recovery effect is taken into account by the model. The rate capacity effect is also considered in this model. For high discharge currents, the charge in the first well is delivered faster to the load in comparison to the charge that flows from the second well. In this scenario there is an amount of charge that remains unused. The consideration of this effect is especially important for applications in electric vehicles, since the unused charge might eventually increase the driving range. The KiBaM yields two difference equations which describe the change of capacity in both wells in dependence of the load i_k, the conductance d and the capacity ratio c:

$$w_{1,k+1} = a_1 w_{1,k} + a_2 w_{2,k} + b_1 i_k, \tag{27}$$

$$w_{2,k+1} = a_3 w_{1,k} + a_4 w_{2,k} + b_2 i_k, \tag{28}$$

where

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} = \mathrm{e}^{\begin{pmatrix} -\frac{d}{c} & \frac{d}{1-c} \\ \frac{d}{c} & -\frac{d}{1-c} \end{pmatrix} \Delta t}$$

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \int_0^{\Delta t} \mathrm{e}^{\begin{pmatrix} -\frac{d}{c} & \frac{d}{1-c} \\ \frac{d}{c} & -\frac{d}{1-c} \end{pmatrix} \vartheta} \mathrm{d}\vartheta \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The term Δt is the sampling time used in the discretization of the model. The battery SOC is given by

$$\mathrm{SOC}_k = \frac{w_{1,k}}{cC_n 3600}, \tag{29}$$

where C_n is the nominal capacity of the battery. The right-hand-side equivalent circuit of Fig. 4 is compounded of three parts, namely, the open circuit voltage V_{OC}, a resistance R_o and two RC networks.

The voltage V_{OC} changes at different SOC levels, as depicted in Fig. 5. The ohmic resistance R_o captures the I-R drop, i.e., the instantaneous voltage drop due to a step load current event. The $R_s C_s$ and $R_l C_l$ networks capture the voltage drops due to the electrochemical and the concentration polarization, respectively. In Fig. 4 the dependency of these parameters on the temperature and on the SOC is represented by the term $(\cdot)$.

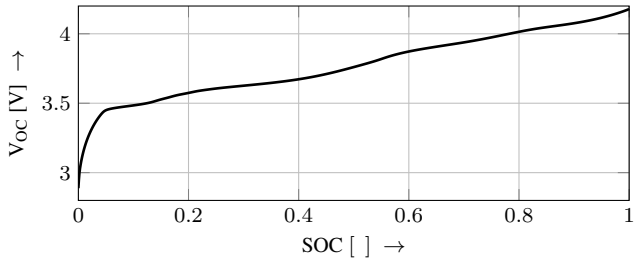


Figure 5. $V_{\mathrm{OC}} - \mathrm{SOC}$ relationship.

This part of the model yields two difference equations which describe the transient response of the battery:

$$v_{s,k+1} = \mathrm{e}^{-\frac{\Delta t}{R_s C_s}} v_{s,k} + \left(-R_s \mathrm{e}^{-\frac{\Delta t}{R_s C_s}} + R_s\right) i_k, \tag{30}$$

$$v_{l,k+1} = \mathrm{e}^{-\frac{\Delta t}{R_l C_l}} v_{l,k} + \left(-R_l \mathrm{e}^{-\frac{\Delta t}{R_l C_l}} + R_l\right) i_k. \tag{31}$$

Accordingly, the state vector of the battery model is given by

$$\mathbf{x}_k = \begin{bmatrix} w_{1,k} & w_{2,k} & v_{s,k} & v_{l,k} \end{bmatrix}^T. \tag{32}$$

The output y_k of the system, represented by the terminal voltage $V_{batt,k}$, is then computed as follows

$$y_k = V_{batt,k}(\mathrm{SOC}) = V_{\mathrm{OC}}(\mathrm{SOC}) + R_o i_k + v_{l,k} + v_{s,k}. \tag{33}$$

5. RESULTS AND DISCUSSIONS

This section evaluates the particle flow particle filter in both accuracy and computational performance in the estimation of the RDR of an electric vehicle. To measure the accuracy of the RDR estimation we employ the relative accuracy (RA) and the alpha-lambda ($\alpha - \lambda$) metric (Saxena, Celaya, Saha, Saha, & Goebel, 2009). In the context of the RDR estimation

the RA is given by

$$\mathrm{RA}_{k_p} = 100\left(1 - \frac{\left|\mathrm{RDR}^*_{k_p} - \mathrm{RDR}_{k_p}\right|}{\mathrm{RDR}^*_{k_p}}\right), \qquad (34)$$

where $\mathrm{RDR}^*_{k_p}$ is the ground truth RDR at time k_p and RDR_{k_p} is the estimated RDR at that time. The $\alpha - \lambda$ metric serves to evaluate whether the estimated RDR lies withing specified bounds.

5.1. Experimental results

The first set of experiments aims to test the suitability of the PFPF in prognostics on the one hand, and to compare its performance in contrast to the PF, on the other hand. To this aim the load profile shown in the top part of Fig.6 is applied to a Li-ion cell until the pre established $\mathrm{SOC}_{\min}$ is reached. For this experiment a cell with a nominal capacity $C_n = 2.15$ Ah, a nominal voltage $V_{\mathrm{nom}} = 4.2$ V and a $\mathrm{SOC}_{\min} = 0.15$ is used. The load profile is computed by scaling down the theoretical load of an electric vehicle driving the standard UDDS (Urban Dynamometer Driving Schedule) drive cycle. In this way it is possible to directly relate the load with the speed of the vehicle and therefore to compute the RDR.

First, the accuracy of the SOC estimation is investigated. To this aim both filters run in parallel and recursively estimate the SOC. The bottom part of Fig.6 depicts the results of the state estimation. As it can be seen, both filters are very accurate while estimating the SOC. The main difference lies on the number of particles used. For the estimation shown just 10 particles are employed by the PFPF, whereas the PF needs 100 in order to estimate the SOC with the same accuracy as the PFPF. This is by no means a claim of improvement of the particle filter for state estimation, but a suggestion that the PFPF successfully manage to estimate states in nonlinear systems with many less particles.

After having proved the applicability of the PFPF for estimating the SOC, the second step is to validate the accuracy and the computational performance of the RDR estimation. To this aim a series of predictions are carried out at different stages of the discharge process every 500 s. Since for this experiment the future load profile of the battery is assumed to be known, the error presented in the RDR estimation is attributed to the model inaccuracy and to the SOC estimation error. A RDR prediction proceeds by simulating the evolution of the battery SOC, from a given time k_p, as a response to the predicted load and by determining the point in the future, at which the $\mathrm{SOC}_{\min}$ is reached. The initial state values at the time of prediction are dictated by the value of the particles obtained from the state estimation step. This procedure is repeated for all particles. The RDR distribution is then computed by means of Eq.(4). As it can be appreciated in Fig.7, the RDR prediction shows a high RA, with the ex-

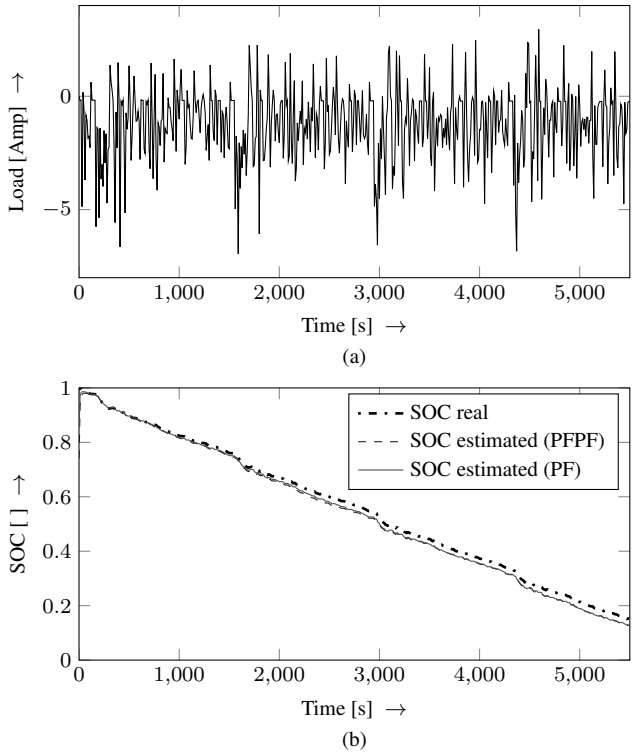


Figure 6. a) Load profile derived from the UDDS drive cycle. b) SOC estimation with the PFPF and the PF.

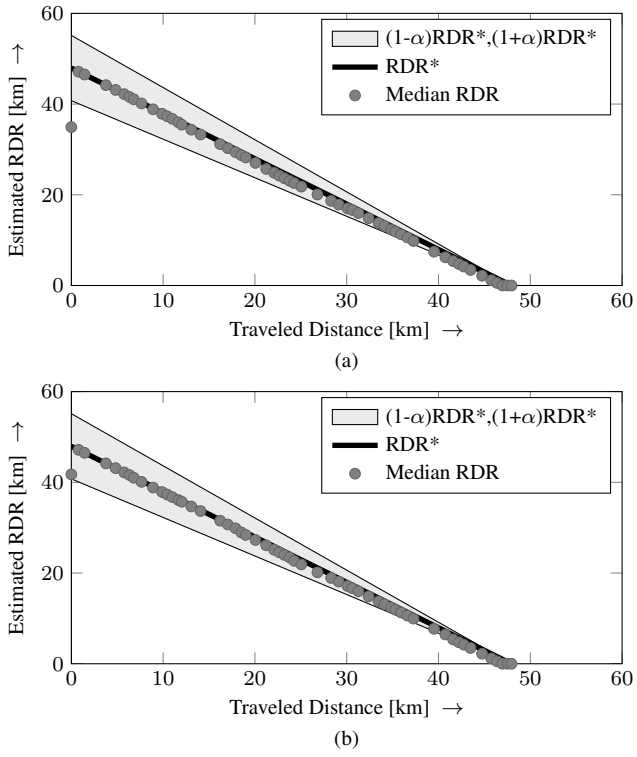


Figure 7. RDR estimation with a) PFPF and b) PF.

ception of the first prediction and the predictions carried out near the end of discharge of the battery. This first deviation is due to the fact that the state estimation in both cases is initialized by uniformly spreading the particles among the entire state space, which causes the estimation to deviate from the real value. Once the filters converge to the real SOC, the RA increases remarkably. As it can be seen, the RA decreases towards the end of discharge at $k_p = 45$ and $k_p = 50$. This phenomenon is attributed to the abrupt voltage drop that the battery exhibits at around $\mathrm{SOC} = 5\%$, as it is shown in Fig. 5. The battery model doesn't accurately capture the behavior of the terminal voltage in this region, which causes the filter algorithm to slightly diverge from the real SOC. Since the uncertainty presented in the filtering step is the only uncertainty considered in this case study, a reduction in the accuracy of the state estimation directly causes a reduction in the RA.

Table 1 presents the RA and the time needed to complete a prediction, here referred as t_{cpu}, for different prediction times. As it can be noted, in average the t_{cpu} of those predictions done with the PFPF are three times faster than those carried out with the PF.

Table 1. RDR prediction performance.

	Urban			
	RA [%]		t_{cpu} [s]	
k_p	PFPF	PF	PFPF	PF
1	72.83	87.05	3.16	3.91
5	100.0	100.0	0.327	1.078
10	99.48	99.48	0.305	0.927
15	98.72	99.44	0.287	0.808
20	97.82	99.33	0.273	0.730
25	96.06	97.22	0.253	0.671
30	95.25	95.67	0.235	0.568
35	95.88	95.98	0.222	0.479
40	94.33	94.53	0.206	0.407
45	88.26	91.41	0.109	0.324
50	76.75	77.80	0.176	0.241

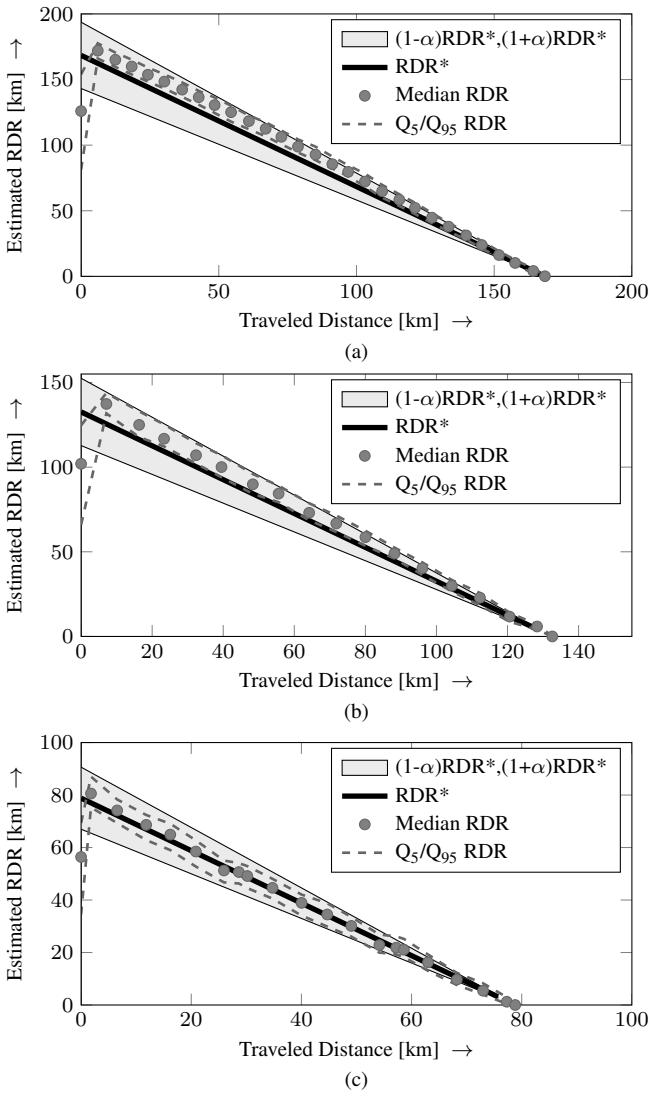


Figure 8. PFPF based RDR estimation in different driving scenarios a) city b) rural areas c) highway.

5.2. Simulation results

A series of simulations is carried out in order to incorporate the uncertainty introduced by the randomness of the driving environment into the RDR estimation. To this aim the methodology previously presented in together with the model of an electric vehicle is used to compute power demand as response to a predicted driving profile, i.e., speed, acceleration and slope profile. The approach for predicting the driving profiles is however out of the scope of this work. The reader is referred to (Oliva et al., 2013) for a detailed explanation about the methodology employed for estimating the RDR.

The RDR prediction proceeds similarly as in the previous section with the difference that in this case each particle is simulated through 50 different predicted driving profiles, i.e, $N_u = 50$. In this case 10 particles are employed by the PFPF and 50 by the PF in order to obtain similar accuracy in the state estimation. As it is shown in Fig.8, the RDR prediction is carried out under three different driving scenarios, namely in the city and rural areas and on the highway.

The simulation results show that the PFPF is also suitable for estimating the RDR even in situations where the future driving load is unknown and that it reduces the computational complexity of the entire prognostics process. In table 2 both the RA and the t_{cpu} for all scenarios is presented. As it can be observed, even though the PF employ more particles than the PFPF, the accuracy in the RDR prediction is in general not better. Furthermore, a noticeable improvement in the computational performance is appreciated in respect to the experimental results. Although, the PF uses just half of the particles as before, the t_{cpu} is now 4 to 5 times larger than the t_{cpu} required by the PFPF in all scenarios.

Table 2. RDR prediction performance under different driving scenarios.

	Driving scenario											
	Urban				Rural				Highway			
	RA [%]		t_{cpu} [s]		RA [%]		t_{cpu} [s]		RA [%]		t_{cpu} [s]	
k_p	PFPF	PF	PFPF	PF	PFPF	PF	PFPF	PF	PFPF	PF	PFPF	PF
1	74.78	88.48	18.21	88.19	76.91	89.00	6.83	23.06	71.56	89.82	3.19	20.50
3	94.19	79.07	16.46	79.83	92.52	87.32	4.06	19.12	97.39	88.00	3.28	17.04
5	93.42	90.54	15.41	72.53	93.42	89.44	3.68	17.03	96.30	87.75	3.10	14.91
7	91.97	90.19	14.64	68.39	93.37	87.13	3.22	15.51	96.91	91.76	2.83	13.79
9	91.80	89.93	13.38	63.40	93.29	88.44	2.88	13.50	98.93	86.91	2.79	13.63
11	89.96	90.37	12.52	57.09	88.52	88.27	2.50	11.46	99.64	94.96	2.64	12.68
13	88.87	91.22	11.37	50.06	89.85	92.74	2.09	9.70	98.73	91.97	2.53	12.18
15	88.60	90.46	10.27	44.70	87.76	77.08	1.81	7.54	98.90	81.45	2.26	11.85
17	88.56	90.59	9.00	38.98	63.20	81.49	1.33	5.39	97.27	74.13	2.11	10.82
21	89.40	89.43	6.65	28.37	–	–	–	–	93.16	81.11	1.88	8.16
25	95.75	89.43	3.88	16.27	–	–	–	–	–	–	–	–

6. CONCLUSIONS AND FUTURE WORK

In this work a methodology for enhancing the computational performance of a particle-filtering-based prognostics approach is presented. The reduction in computational complexity is achieved by reducing the number of particles needed in the state estimation and thereby reducing the number of simulations needed to determine the RUL of the system. The reduction of particles is carried out by applying a deterministic flow, which migrates the particles through the state space in an optimal manner from the prior to the posterior state estimate. The advantage of such a migration allows us to employ less particles in contrast to the standard particle filter, since the particles are moved to the correct location obeying to the Bayes's rule. Such a particle reduction is highlighted during the prediction step, due to less simulations are needed for determining the distribution of the RUL.

The proposed methodology is afterwards illustrated and validated by means of the RDR estimation problem, in which is desired to determine the distance that can be driven by an electric vehicle with the energy stored in the battery pack at given points in time. Both experimental and simulation results show that the particle flow particle filter successfully reduces the computational burden associated with the estimation of the RUL in nonlinear systems.

Even though the presented approach exhibits both good computational performance and estimation accuracy, it is worth mentioning that the experiments carried out are based just on state estimation. That is, no joint or dual state/parameter estimation is done. This is justified by the assumption that the parameters of the battery model degrade very slow within the time span of a trip. However, a more proper implementation of the RDR estimation problem requires estimating the parameters together with the states in order to account for the aging effect of the battery. We therefore aim to investigate in the future the applicability and performance of the particle flow particle filter for a joint state/parameter estimation.

ACKNOWLEDGMENT

The funding for this work was provided by the EU and the federal state of North Rhine-Westphalia (NRW) in frame of the Ziel2 project "Technology and test platform for a competence center for interoperable electromobility, infrastructure and networks" (TIE-IN).

REFERENCES

Daigle, M., & Goebel, K. (2010). Improving computational efficiency of prediciton in model-based prognostics using the unscented transform. In *Annual conference of the prognostics and health management society 2010.*

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society 2012.*

Daum, F., & Huang, J. (2008). Particle flow for nonlinear filters with log-homotopy. In *Proceedings of spie conference* (Vol. 6969).

Daum, F., & Huang, J. (2010). Exact particle flow for nonlinear filters: Seventeen dubious solutions to a first order linear underdetermined PDE. In *Signal processing, sensor fusion, and target recognition XXII* (p. 64-71).

Daum, F., & Huang, J. (2011). Particle degeneracy: root cause and solution. In *Proceedings of spie conference* (Vol. 8050).

Daum, F., & Huang, J. (2013). Particle flow with non-zero diffusion for nonlinear filters. In *Proceedings of spie conference* (Vol. 8745).

George, A., & Powell, W. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. In *Journal of machine learning* (Vol. 65, p. 167-198). Kluwer Academic Publishers.

Jongerden, M., & Haverkort, B. (2009). Which battery model to use? In *Software, IET* (Vol. 15, p. 445-457).

Julier, S., & Uhlmann, J. (2004). Unscented filtering and

nonlinear estimation. In *Proceedings of the IEEE.*

Manwell, J. F., & McGowan, J. G. (1994). Extension fo the kinetic battery model for wind-hybrid power systems. In *Proceedings of EWEC.*

Oliva, J. A., Weihrauch, C., & Bertram, T. (2013). A model-based approach for predicting the remaining driving range in electric vehicles. In *Annual conference of the prognostics and health management society 2013.* (p. 438-448).

Orchard, M., & Vachtsevanos, G. (2010). A particle-filtering approach for on-line fault diagnosis and failure prognosis. In *Transactions of the institute of measurement and control* (Vol. 31, p. 221-246).

Restaino, R., & Zamboni, W. (2013). Rao-blackwellised particle filter for battery state-of-charge and parameters estimation. In *Industrial electronics society, iecon 2013 - 39th annual conference of the ieee* (p. 6783-6788).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostics performance metrics. In *Annual conference of the prognostics and health management society 2009.*

Tsiligkaridis, T., & Hero, A. (2013). Covariance estimation in high dimensions via kronecker product expansions. In *Signal processing, ieee transactions on* (Vol. 61, p. 5347-5360).

BIOGRAPHIES

Javier A. Oliva received his B.S. degree in Mechanical Engineering from the University Landivar in Guatemala in 2006 and his M.S. degree in Automation and Robotics from the Technische Universität Dortmund in 2010. His research interests include probabilistic methods, diagnosis and prognostics applied to electric vehicles. He is currently working as researcher at the Institute of Control Theory and Systems Engineering from the TU Dortmund in the area of driver assistance systems for electric vehicles.

Torsten Bertram is Professor at the Technische Universität Dortmund and he directs the Institute of Control Theory and Systems Engineering. He has carried out applied research in the areas of drive systems, service robotics and development methodology.

Efficient Dependency Computation for Dynamic Hybrid Bayesian Network in On-line System Health Management Applications

Chonlagarn Iamsumang, Ali Mosleh, Mohammad Modarres

The Center for Risk and Reliability
University of Maryland College Park, Maryland, USA
kci@umd.edu, mosleh@umd.edu, modarres@umd.edu

ABSTRACT

This paper presents a new dependency computational algorithm for reliability inference with dynamic hybrid Bayesian network. It features a component-based algorithm and structure to represent complex engineering systems characterized by discrete functional states (including degraded states), and models of underlying physics of failure, with continuous variables. The methodology is designed to be flexible and intuitive, and scalable from small localized functionality to large complex dynamic systems. Markov Chain Monte Carlo (MCMC) inference is optimized using pre-computation and dynamic programming for real-time monitoring of system health. The scope of this research includes new modeling approach, computation algorithm, and an example application for on-line System Health Management.

1. INTRODUCTION

With increasing complexity of today's engineering systems that contain various component dependencies and degradation behaviors, there has been increasing interest in real-time System Health Management (SHM) capability to continuously monitor sensors, software, and hardware components for detection and diagnostic of safety-critical systems. The modeling framework should be flexible to accommodate the complexity of component dependencies and failure behaviors, such as sequence-dependent failures, functional dependencies, etc.

Bayesian Network (BN) (Pearl, 1986) (Jensen, 2001) and their extension for time-series modeling known as Dynamic Bayesian Network (DBN) (Friedman, 1998) (Murphy, 2002) have been shown by recent studies to be capable of providing a unified framework for system health diagnosis and prognosis (Ferreiro, Arnaiz, Sierra, & Irigoien, 2011) (Tobon-Mejia, Medjaher, Zerhouni, & Tripot, 2012) (Schumann, Rozier, Reinbacher, Mengshoel, Mbaya, & Ippolito, 2013). Bayesian Network has many modeling features, such as multi-state variables, noisy gates, dependent failures, and general posterior analysis (Wilson & Huzurbazar, 2007) (Langseth & Portinale, 2007) (Doguc & Ramirez-Marquez, 2009). It also allows a compact representation of the temporal and functional dependencies among system components (Boudali & Dugan, 2006) (Weber & Jouffe, 2006).

The main advantage of using BN in system reliability is its simplicity to represent systems and the efficiency for obtaining component associations. Another important benefit of BNs is that they enable us to integrate information from different sources, including experimental data, historical data, and prior expert opinion. This feature is particularly useful for the reliability assessment of fault tolerant systems, where failure data from tests and field operations are sparse and obtained from diverse source of information. Bayesian networks are particularly well suited to modeling systems that we need to monitor, diagnose, and make predictions about, all under the presence of uncertainty.

However, one of the barriers to applying BN to real-world problems is to be able to adequately handle the "hybrid models", which contain both discrete and continuous variables with general static and time-dependent failure distributions. Despite the advances in BN researches, the previous applications of BNs as mainstream technology for SHM problems remain modest. To date, the BN framework has only partially addressed these limitations (Lauritzen & Jensen, 2001) (Moral, Rumi, & Salmeron, 2001) (Lerner, 2002) (Shenoy, 2006). The vast majority of BNs used in real world applications are either purely discrete or purely continuous.

For hybrid BNs containing mixtures of discrete and continuous nodes with non-Gaussian distributions, exact inference becomes computationally intractable (Boyen & Koller, 1998). The common approach to handling (non-

Chonlagarn Iamsumang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Gaussian) continuous nodes is to discretize them using some pre-defined range and intervals (Neil, Tailor, Marquez, Fenton, & Hear, 2007). This is cumbersome, error prone and usually inaccurate.

Even though a universal framework for hybrid BN is currently impracticable, a special case algorithm can be effective in SHM where a relatively small subset of possible values covers a large proportion of all possible values typically encountered. This paper presents a hybrid BN-based methodology for component degradation model and efficient algorithms to apply them in online health monitoring of complex systems.

The focus of this research is to enable probabilistic diagnosis and prognosis of system in real-time by optimizing Markov Chain Monte Carlo inference with pre-computation and dynamic programming to reduce the computation time and number of inferences required. Efficient computation allows on-line system monitoring and provides on-demand system health inquiry for operators to make maintenance decision and to prioritize which part of the system to investigate to avoid an accident.

2. PROPOSED METHODOLOGY

2.1. Hybrid Bayesian Network

For SHM modeling, it is advantageous and intuitive to consider a hybrid system, typically with the continuous variables being modeled as continuous and the system's functionality probability being discrete.

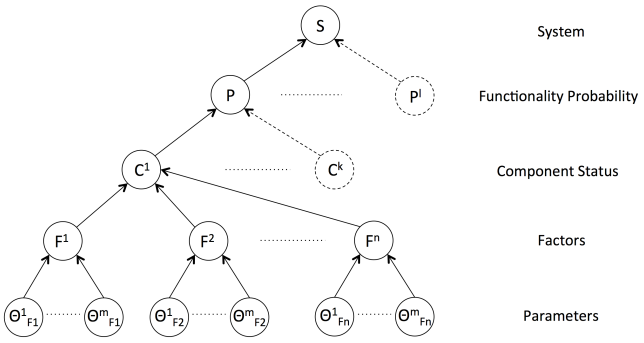


Figure 1: Overview of different levels in SHM Bayesian Network

The proposed complex system hybrid BN can be separated into 5 levels as shown in Figure 1, according to the typical characteristics of the nodes. The BN combines high-level functionality nodes with low-level physical of failure nodes. Here are the descriptions of each level:

1. System node: this is the highest level of nodes with no children. It represents the state of the whole system and usually indicates whether or not the system is working as intended.
2. Functionality probability nodes: these nodes are designed to be abstract discrete nodes that represent various functionalities, which are required for the system to operate.
3. Component status nodes: these are continuous nodes representing states of physical components susceptible to specific failure mechanisms in the system. These values should be measurable directly or indirectly.
4. Factor nodes: these nodes contribute to the degradation of the components. They can be component internal factors related to material properties or physical characters, or they can be external factors such as environmental stress or temperature.
5. Parameter nodes: these nodes are hyper-parameters that describe probability distributions of the factors.

It is to be noted that each level does not have to be only one layer as shown in Figure 1, it can be a combination of different layers of nodes that have the same type.

Reliability concerns arise when some critically important materials or devices degrade with time. Let C represent a critically important material/device parameter. This parameter degrades over the life of the component. The value itself can either increase (threshold voltage of a semiconductor device, increase in leakage of a capacitor, increase in resistance of a conductor) or decrease (decrease of pressure in a vessel, decrease of spacing between mechanical components, decrease in lubricating properties of a fluid). Figure 2 presents the SHM BN at a specific time, t. The shaded areas show continuous nodes that are related to each component.

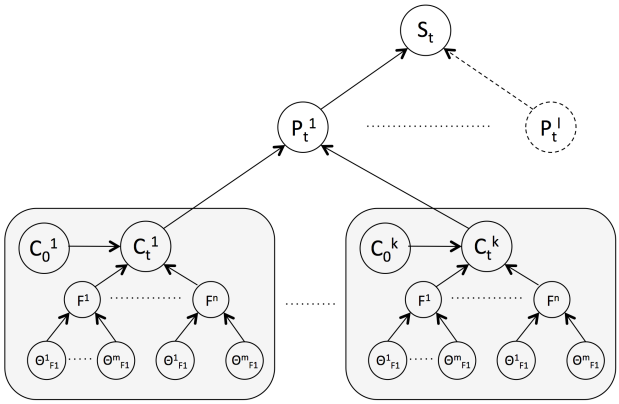


Figure 2: SHM Bayesian network at specific time t.

A Taylor expansion about t=0 produces the Maclaurin Series, assuming that C changes monotonically and relatively slowly over the lifetime of the material/device:

$$C(t) = C_{t=0} + \left(\frac{\partial C}{\partial t}\right)_{t=0} t + \frac{1}{2}\left(\frac{\partial^2 C}{\partial t^2}\right)_{t=0} t^2 + \cdots \qquad (1)$$

By assuming that the higher order terms in the expansion can be approximated by simply modeling degradation of component/device parameter *C* with a power-law equation:

$$C = C_0[1 \pm A_0 t^m] \quad (2)$$

Where C_0 is the value of C at $t = 0$, A_0 is material/device-dependent coefficient, and m is the power-law exponent. Both A_0 and m are parameters that can be learned from component/device degradation data. Summation (+) is used when the parameter C increases with time, while subtraction (-) is used when the parameter C decreases with time.

A_0 is generally material/microstructure dependent. It is not only a function of material variations, but also a function of other factors, such electrical, thermal, mechanical and chemical environments to which the device is exposed.

$$A_0 = A_0(F_1, \dots, F_n) \quad (3)$$

Therefore, we have:

$$C = C_0[1 \pm A_0(F_1, \dots, F_n) t^m] \quad (4)$$

m and other parameters are considered to be constant for the component/device. Considering a Bayesian network at a time slice of a given system, t is then constant and indicates the current life of the component/device.

For a component/device to fail, the amount of degradation must reach a critical value, C_{crit}. Therefore, the time to failure, $T_{failure}$, is then:

$$T_{failure} = \left[\frac{1}{\pm A_0(F_1, \dots, F_n)}\left(\frac{C_{crit} - C_0}{C_0}\right)\right]^{1/m} \quad (5)$$

Since the component parameter and their parents are continuous nodes, and the functionality probability nodes are discrete, the interface between these different types of nodes becomes critical. In general hybrid BNs, when continuous nodes have discrete parents, there are simple conditional inference techniques such as in conditional linear Gaussian (CLG) model. Difficulty arises when discrete nodes have continuous parents, which is the case for our SHM network. However in this case, even though discrete functionality probability nodes have continuous component status nodes, they are related by degradation thresholds.

Discrete functionality nodes can contain more than 2 states with thresholds between the transitions of one state to the other. Let the threshold value between functionality state i and j be $C_{th,i/j}$. The most common case would be state i denotes the component function, and state j denotes the component does not function. Let P_i be the probability of functionality being in state i. The probability P_i is then the probability that the component status C is lower than the threshold value $C_{th,i/j}$. Figure 3 shows a typical component exponential degradation function and the overlap of probability distributions of C and $C_{th,i/j}$.

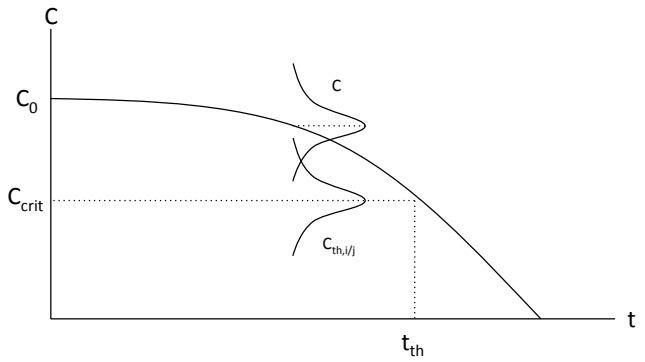


Figure 3: Overlap of probability distribution of component status and its threshold.

Let a functionality node has n states, the probabilities of being in the states are $P_1, \dots, P_n$. Assume the state of the functionality node changes monotonically according to the component degradation status:

$$C_{th,i-1/i} < C_{th,i/i+1} \text{ for } i = 2, \dots, n-1 \quad (6)$$

Therefore,

$$P_i = prob\left(C_{th,i-1/i} < C < C_{th,i/i+1}\right) \quad (7)$$

Analytically, P_i can be calculated from the following convolution equation:

$$P_i = \int_{-\infty}^{C_{th,i/i+1}} \int_{C_{th,i-1/i}}^{\infty} \int_{C_{th,i-1/i}}^{C_{th,i/i+1}} p\left(C_{th,i-1/i}\right) \cdot p(C) \cdot p\left(C_{th,i/i+1}\right) dC dC_{th,i/i+1} dC_{th,i-1/i} \quad (8)$$

If there are many component critical parameters contribute to this functionality then the state of the functionality node conditionally depends on comparison between the status of each component and its threshold values.

2.2. Dynamic Bayesian Network

Dynamic Bayesian Network (DBN) is a Bayesian network that includes a temporal dimension. This new dimension is managed by time-indexed random value t to indicate time stage of the nodes. A set of nodes at certain stage contains random variables relative to time slice t. An arc that links two variables belonging to different time slices represents a temporal probabilistic dependence between these variables. Variables can be modeled to have impact on the future distribution of the other variables. These impacts are defined as transition probabilities between the stats of variables at time step t and $t + \Delta t$.

A DBN describes the joint distribution of a set of variables **θ.** This is a complex distribution, but may be simplified by using the Markov assumption. The Markov assumption requires only the present state of the variables $\boldsymbol{\theta}_t$ to estimate $\boldsymbol{\theta}_{t+1}$, i.e. $p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_t) = p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$ where p indicates a probability density function and bold letters indicate a vector quantity. Additionally, the process is assumed to be stationary, meaning that $p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$ is independent of t.

For SHM Bayesian network, the main variables that change between time slices are component parameters. Components degrades over time, therefore, the status of components at a certain time slice depend on their status at the previous time slice and the factors affecting the degradation processes during that transition.

$$p(C_t) = p(C|C_{t-\Delta t}, \{F_t^1, \ldots, F_t^n\}) \tag{9}$$

Given that F_t^i is the average value of factor i between time slice $t - \Delta t$ and t.

Figure 4 shows a two-time-slice representation of a dynamic SHM Bayesian network. Δt should be set according to the system under interest and how often the parameters can be observed, such as frequency of sensor signals.

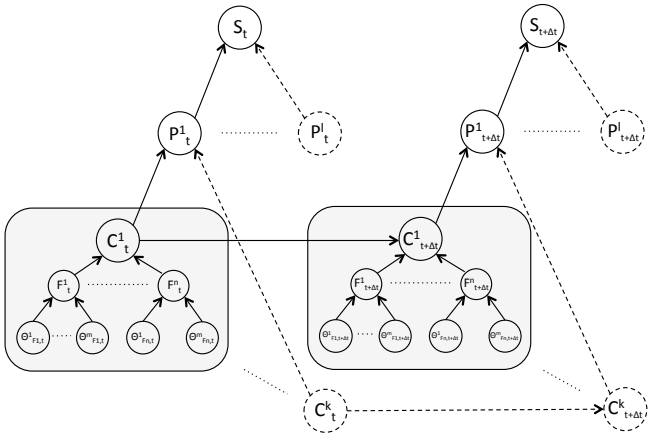


Figure 4: Two-time-slice representation of a dynamic SHM Bayesian network

At any point in time during system operation, any value of variables in the system can be derived by probabilistic inference to compare with its expected value to see if the probability is still in the acceptable range and the system as a whole is working as intended. With continuous monitoring, the trajectory of the degradation processes can be estimated form our knowledge of the health of the system. We can then use this information to estimate remaining useful life (RUL) of components and plan maintenance accordingly.

2.3. Inference

Bayesian network is a complete model for the variables and their relationships. Therefore, it can be used to answer probabilistic queries about them. The main application is to use BN to realize updated knowledge of the states of a subset of variables, when the other variables (the evidence variables) are observed.

Bayes' rule with continuous variables:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{\int d\theta\, p(D|\theta)p(\theta)} \tag{10}$$

Let θ be a parameter value and D is data value of the evidence, $p(\theta|D)$ is then the posterior probability of getting parameter value θ when data value D is presented.

In real world SHM applications, there are various types of parameter distributions, which make it difficult to calculate full marginal distributions analytically. Therefore, sampling techniques can be used to approximate the distributions instead. Expected values of a distribution can be estimated as follow:

$$E[p(\theta|D)] \approx \frac{1}{N}\sum_{n=1}^{N} p\left(\theta^{(n)}|D\right) \tag{11}$$

Where $\theta^{(1)}, \ldots, \theta^{(n)}$ are the sample values of parameter θ.

There are many ways to sample these values, the key idea is to let θ values be points in state space and find a way to walk around so that the likelihood of visiting any point θ is proportional to $p(\theta)$. Therefore, the sampler will spend more time sampling from the distribution where the probability is high, and spending less time sampling from where the probability is low. This can be achieved by using Markov chain Monte Carlo (MCMC) algorithm (Cousins, Chena, & Frisse, 1993) (Dagum & Horvitz, 1993).

MCMC algorithms produce random walks over a probability distribution. By taking a sufficient number of steps in this random walk, the MCMC simulation algorithm visits various regions of the parameter space in proportion to their posterior probabilities. We can, for inferential purposes, summarize the iterates obtained in these random walks much as we would summarize an independent sample from the posterior distribution.

The procedure for updating the belief about the system state as new information becomes available is called Bayesian recursive filtering.

$$p(\theta_t|D_{1:t}) = \frac{p(D_t|\theta_t)p(\theta_t|D_{1:t-1})}{\int d\theta\, p(D_t|\theta_t)p(\theta_t|D_{1:t-1})} \tag{12}$$

Under certain assumptions, such as when the system is linear Gaussian, the belief state will be of a known parametric form and computationally efficient solutions to the filtering problem (e.g. Kalman filter, extended Kalman filter, unscented Kalman filter) are available. Outside such assumptions, a computationally feasible method for

inference in the DBN is particle filtering, a form of sequential Monte Carlo based on Bayesian recursive filtering. Common particle filtering methods are based on sequential importance sampling (SIS) (Chen, 2003).

3. COMPUTATIONAL ALGORITHM

In highly complex systems, MCMC algorithm requires large amount of computational time for inference in hybrid DBN. The computation time grows exponentially with each additional layer of network and becomes infeasible with large number nodes. The computation time makes it impossible for on-line health monitoring of complex systems. To solve this problem, special case algorithm for SHM is introduced to reduce the number of computations and the amount of time required for each computation.

One of the main characteristics of SHM in contrast of other applications is that during a normal operation, the environmental factors that affect component degradation process are expected to be roughly the same and predictable. Therefore, instead of performing Bayesian updating at a specific time interval, it only needs to be done when a factor value changes outside of expected range.

$$|f_t - f_{t-1}| > \epsilon_f \tag{13}$$

Where ϵ_f depends on the sensitivity of component status due to the change in value of that factor. Please note that this is possible because component status is a function of time. Therefore, the degradation of a component between time period t_i to t_j where the change in factor value is less than ϵ_f will take a normal distribution $\mathcal{N}(\mu_f, \sigma_f)$ for $\Delta t = t_j - t_i$.

3.1. Pre-computation

Since the values are predicted to be in certain ranges, it is possible to perform pre-computation for all combinations of possible values in the ranges before the system is in operation. The results are then stored in a database, such that they can be pulled quickly to approximate the inferences in real-time. More computation should be conducted and more results should be added to the database as the health of the system is being monitored such that the database will cover all the possible computations that may be needed in the future.

With continuous range of parameter values, it is impossible to pre-compute every possible outcome. The goal of pre-computation is to cover enough values of observable parameters, so that the values of unobservable parameters can be accurately interpolated from the results.

There are two factors in considering the selection of possible values.

First is the range of observable parameters after a time period Δt. The selections should cover full range of possible values. There should be at least one selected value at lower bound and one selected value at upper bound. The common range is from 5th percentile to 95th percentile, or more accurately 0.5th percentile to 99.5th percentile.

Second is the number of selections within the bound: the higher the number of selections, the more accurate results from interpolation will be. The density of selections should be proportional to the probabilistic density of the observable parameters. For example, if there is N number of selections per variable, the selections are:

$$C_s = \left\{C^{p_{low}th}, C^{p_{low}+\delta th}, C^{p_{low}+2\delta th}, \dots C^{p_{high}th}\right\} \tag{14}$$

$$\delta = \frac{p_{high} - p_{low}}{N_{selections} - 1} \tag{15}$$

Therefore, for a given measurement interval Δt, we can estimate the set of possible values and use those values to pre-computed possible outcomes.

There are two different types of observable parameters. The first one is the parameters that change over time. This is usually the case for component status parameters. For pre-computation to be feasible, the changes must be predictable. For a component status parameter, the change in value can be computed from its degradation equation for a given Δt. Figure 5 shows example expected value, 5th percentile, and 95th percentile values.

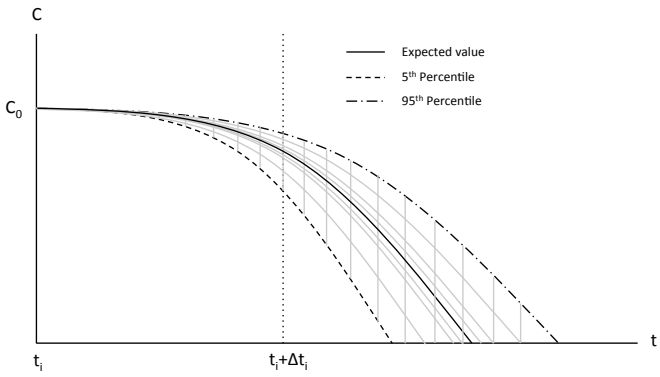


Figure 5: Example component status degradation with 5th percentile, and 95th percentile values.

For this case, the range of possible values grows over time. Therefore, the number of selections should increase proportionally with the range to keep the interval between selected values the same, thus, keep the accuracy of interpolation constant.

The other type of observable parameters is constant parameters. These parameters are usually Gaussian distributed. For this case, the range always stay constant, therefore, the selections remain the same throughout the life of the component.

One advantage of the isolation among component sub-tree is that time intervals do not have to be uniform for all

components. Measurement/inspection intervals can be based on the rate of component degradation and possible change to component parameters. They can also be dynamically changed during the life a component depending on its status.

For example there can be less frequency of measurements during the early life of a component due to less probability of failure. Then increase the frequency when the component approaches the end of life.

$$\Delta t \propto \frac{1}{\Delta C} \tag{16}$$

The time interval between measurements, Δt, should then be inverse proportional to the amount of change of the parameter C. Therefore, the sampling rate around a certain evidence value will be proportional to the probability that the evidence value could happen and how much different in values to the possible values around it at certain period of time.

If the observed values are always in the predicted range, the accuracy of the results depends upon the number of selections for pre-computation. The number of selections is the number of selections at each time-slice multiplies be the number of measurement intervals. The number of pre-computations is then the number selections for each observable times the number of observables parameters.

$$N_{pre-computation} = \sum_{j=0}^{T_c/\Delta t} \left(\prod_{i=1}^{n} N_{selections,i,t+(j\Delta t)} \right) \tag{17}$$

Where $N_{selections,i,t}$ is the number of selections of observable parameter i at time t. n is the number of observable parameters. T_c is the component life.

The total computation time then can be estimated.

$$T_{pre-comp} = N_{pre-comp} \cdot T_{average-per-comp} \tag{18}$$

For MCMC computation, the average computation time is proportional to the number iterations. The higher the number of iterations, the higher accuracy of the result will be. Therefore, there is a tradeoff between computation time and accuracy. For pre-computation, the decision between higher number of value selections or higher number of iteration per computation must be made.

3.2. Dynamic Programming

Dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. It is applicable to problems exhibiting the properties of overlapping subproblems and optimal substructure. When applicable, the method takes far less time than naive methods that don't take advantage of the subproblem overlap.

In general, to solve a given problem, we need to solve different parts of the problem (subproblems), then combine the solutions of the subproblems to reach an overall solution. Often when using a more naive method, many of the subproblems are generated and solved many times. The dynamic programming approach seeks to solve each subproblem only once, thus reducing the number of computations: once the solution to a given subproblem has been computed, it is stored the next time the same solution is needed, it is simply looked up. This approach is especially useful when the number of repeating subproblems grows exponentially as a function of the size of the input.

Using dynamic programming can reduce the pre-computation time for Bayesian Network inference drastically. Instead of computing full inferences for each set of evidence values, dynamic programming algorithm retain marginal results that can be reused with similar set of evidence values.

There are three steps for the algorithm. First, use logic-sampling algorithm and degradation model to generate all possible evidence values according to its probability of occurring. Not all evidence nodes have to be instantiated for each case, only the evidence nodes that are required for observing nodes are instantiated.

Second, check and construct a cache by comparing each generated case to those already in the cache. If the case is found to be new, this algorithm determines, the joint probability of the case's evidence using the algorithm in the third step.

Third, the marginal posterior-probability distributions over the diagnosis nods are determined, then the values of the evidence nodes, the joint probability of the evidence set, and the marginal posterior-probability distributions for the diagnosis node are stored in the cache.

Figure 6 shows two example cases where dynamic programming can reduce the number of computation. The first case is when nodes have the same set of parent nodes, thus the same sets of possible marginal probability distributions for discrete nodes. The second case is when continuous parameters have several trajectories that can reach the same values after some period of time.

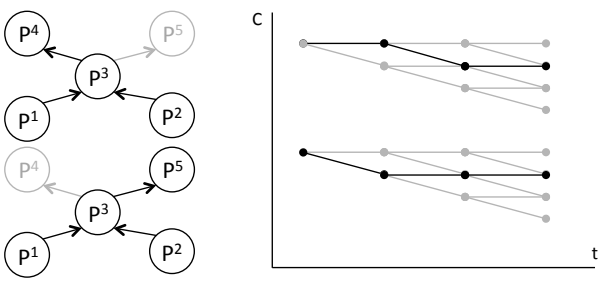


Figure 6: Example cases where dynamic programming reduces number of computations

In addition, if more computations are needed during an operation in the event where evidence values reaches the bound of expected values, dynamic programming provide a set of marginal results that can be used for possible faster inference of values outside the pre-computed cache.

Since both deterministic and approximate inference were found to be NP-hard (Cooper, 1990) (Dagum & Luby, 1993), the computation complexity for both discrete functionality and continuous component degradation model are exponential in the network's treewidth. Figure 7 shows a plot presenting differences between pre-computation time with and without dynamic programming. Without storing marginal probability distribution results for further computations, all approximate inference computations are required for pre-computation, thus increases computation time exponentially with network's treewidth.

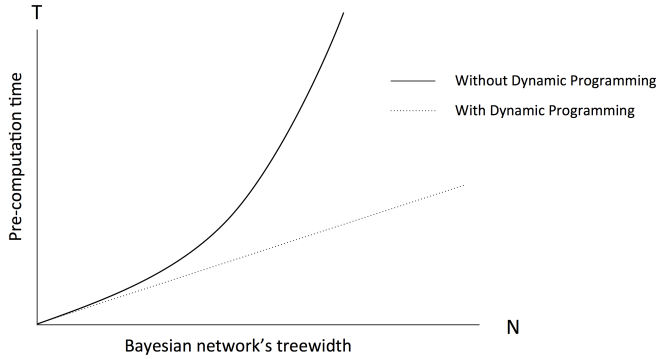


Figure 7: Inference pre-computation time with and without dynamic programming.

3.3. Efficient Dependency Algorithm

In the case that components in the system are dependent to each other because there have common factors, an efficient algorithm is required to maintain the proposed modular component model.

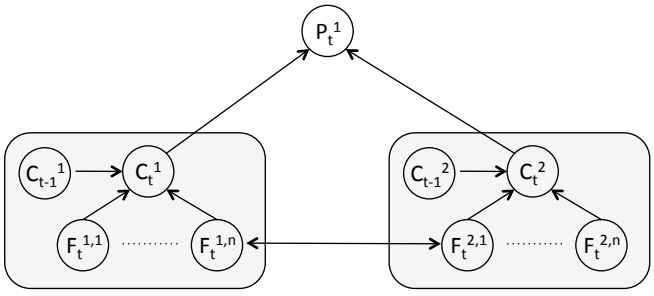


Figure 8: BN of components with common factor

Figure 8 shows an example of 2 components system where both component shares the same factor ($F_t^{1,n}$ and $F_t^{2,1}$). The common approach is to combine the nodes, however, this method is not ideal due to the following reasons:

First, even though the components share the same factor, there is very likely a spatial different between the two components. By combining the nodes, the possibility of decoupling them is eliminated from future analysis. For example, two components are directly in contact of each other and they are assumed to always have the same temperature. There is a chance that in some scenarios, the two components are separated due to an external event or unexpected degradation, the model should be flexible enough to handle this situation.

Second, combining the nodes makes the model no longer modular. Continuous variables inference cannot be done within the component sub-system. This leads to huge increase in complexity and computation time.

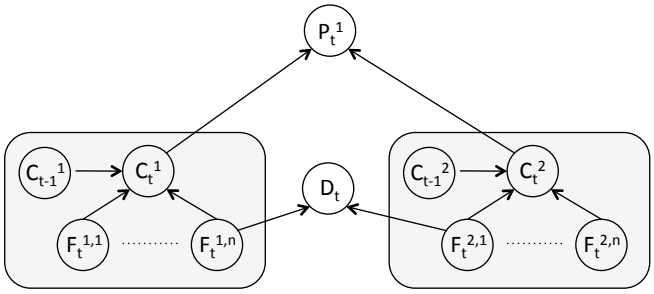


Figure 9: Proposed BN with observable common factor

Let D_t be a node representing the value observed from a detector/sensor used to measure a common factor. If the common factor is observable, factor $F_t^{1,n}$ and $F_t^{2,1}$ can be directly derived from the measurement value of D_t. Therefore, inference calculations for each component stay modular. Figure 9 shows the proposed BN when the common factor is observable.

$$p(C_t) = p(C|C_{t-\Delta t}, \{F_t^1, \dots, F_t^n\}) \tag{19}$$

$$p\left(F_t^{1,n}\right) = p\left(F_t^{1,n}|D_t\right) \tag{20}$$

If the common factor is unobservable, the inference calculation can be done by placing a hidden node D_t as an imaginary measurement node between $F_t^{1,n}$ and $F_t^{2,1}$, shown in Figure 10.

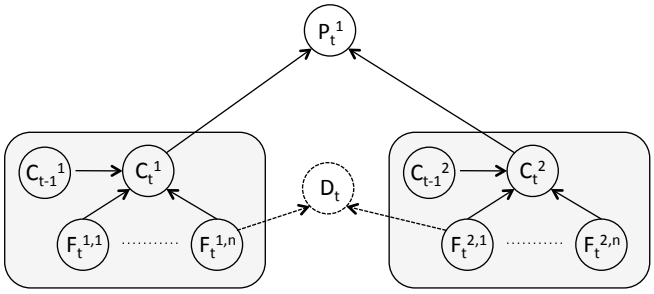


Figure 10: Proposed BN with unobservable common factor

Since we know that $F_t^{1,n}$ and $F_t^{2,1}$ are more likely to have the same value, $p(F_t^{1,n},F_t^{2,1})$ is expected to have a distribution similar to Figure 11.

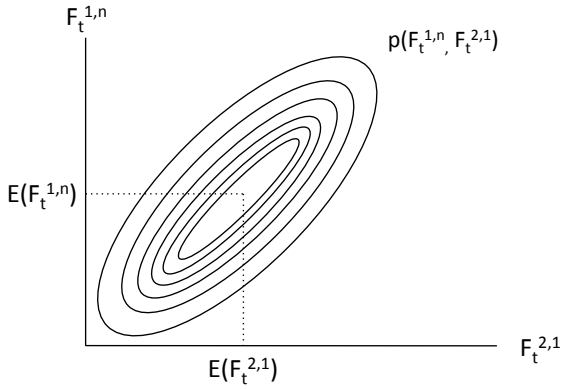


Figure 11: Probability distribution of the common factor.

One method to reduce computation complexity and keep the inference calculation modular is to incorporate pre-computation approximation. Pre-computation generates possible subsets of values of variables according to their probability distribution. For this case:

$$for\ p(P_t^1|C_t^1,C_t^2)\sim\forall i,j\ p\{P_t^1|(C_t^1)_i,(C_t^2)_j\} \quad (21)$$

Therefore, the combination of $\{(C_t^1)_i,(C_t^2)_j\}$ that have higher probability are the ones that the values of $F_t^{1,n}$ and $F_t^{2,1}$ are similar.

$$p\left(\{(C_t^1)_i,(C_t^2)_j\}|F_t^{1,n}\approx F_t^{2,1}\right)>p\left(\{(C_t^1)_i,(C_t^2)_j\}|F_t^{1,n}\not\approx F_t^{2,1}\right) \quad (22)$$

Using this method, the most probable explanation (MPE) can be derived in real-time from the pre-computation cache.

In summary, this section presented new comprehensive computational algorithms that support the proposed SHM model with dependency between components. The combination of pre-computation and dynamic programming techniques is shown to be a feasible method for real-time system-wide inferences in complex hybrid BN.

4. EXAMPLE APPLICATION

Consider integrated circuits (ICs) with both electromigration (EM) and stress migration (SM) degradations. Let $C^{(1)}$ and $C^{(2)}$ be component status degrading under EM and SM respectively.

$$C^{EM}=C_0{}^{EM}\left[1-A_0{}^{EM}\left(J^{(e)}-J_{crit}^{(e)}\right)^{r^{EM}}\exp\left(\frac{-Q^{EM}}{K_BT}\right)t^{m^{EM}}\right] \quad (23)$$

$$C^{SM}=C_0{}^{SM}\left[1-A_0{}^{SM}(L)^{r^{SM}}\exp\left(\frac{-Q^{SM}}{K_BT}\right)t^{m^{SM}}\right] \quad (24)$$

$J^{(e)}$ is the electron current density. $J_{crit}^{(e)}$ is a critical (threshold) current density which must be exceeded before significant EM is expected. L is the tensile stress in the metal for a constant strain.

The BN model of a component affected by EM and SM is shown in Figure 12.

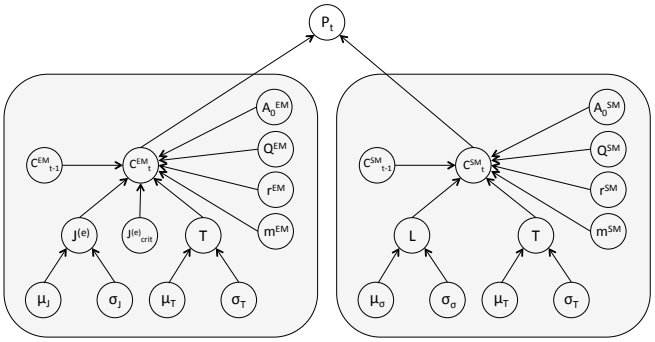


Figure 12: BN of a component with EM and SM degradations

Assume $J^{(e)}$, L, and T are expected to be normally distributed between time *t-1* to *t*,

$$J^{(e)}=\mathcal{N}(\mu_J,\sigma_J),L=(\mu_L,\sigma_L),T=\mathcal{N}(\mu_T,\sigma_T) \quad (25)$$

In the context of simple health monitoring in this example, $A_0, Q, r,$ and m are considered to be constant parameters representing material/device internal factors. These parameters can also be modeled with probabilistic distributions.

Consider an Al-alloy under high temperature operation, with current density J = 2×10^6 A/cm^2 and at a metal temperature T = 200 °C. Assume an activation energy of Q = 0.8 eV for electromigration and 0.6eV for grain boundary diffusion.

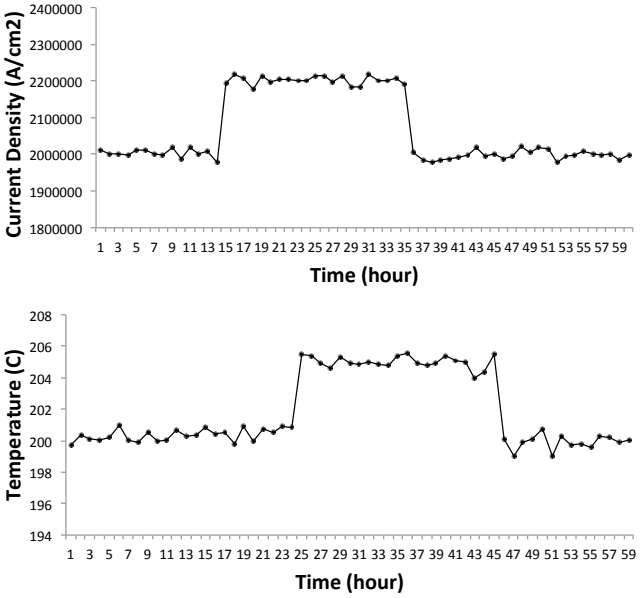


Figure 13: Current density and temperature data set

The current density exponent of n = 2 and stress migration exponent of n=2. Using conservative design approach, assume $J_{crit} = 0$.

The data of current density and temperature, show in Figure 13, is retrieved once per hour during 60 hours of operation. Figure 14 shows an example plot of component degradation under electromigration vs. time at different current density and temperature, including from the data set. Approximate inference of component parameter is available almost instantly with pre-computation of C_t at t = 1,…,60, with the range of J between 1.8×10^6 A/cm^2 to 2.2×10^6 A/cm^2, and T between 90°C to 120°C.

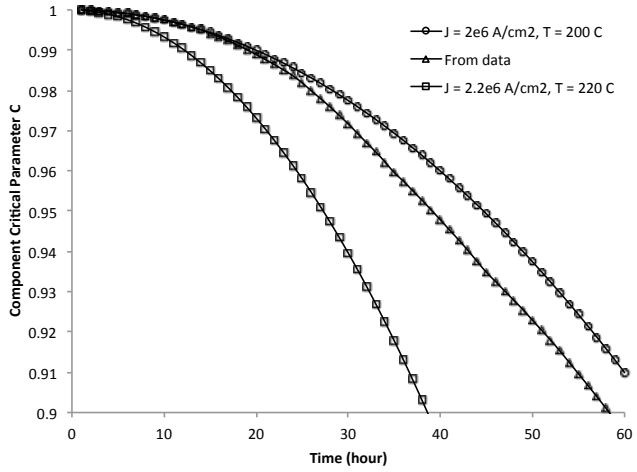


Figure 14: Plot of component parameter C^{EM} vs. time at different J and T, including from the data set

With traditional BN modeling, both failure modes have temperature as a common factor. Therefore, the component parameters, CEM and CSM, have the same parent node, T. In this case, any approximate inference will require full marginal distribution of both failure mode variables. The amount of time for sampling and computation increases exponentially with the number of variables in the inference calculation. With the proposed technique, the failure modes stay modular and approximate inferences can be achieved at much lower cost because of lower number of variables in the calculation. For this example, approximate inference calculation will only involve parameters of failure mode EM and failure mode SM, but not both of them combined.

This also allows real-time inquiry of the states of degradations of all components in the system without computing full inference of all nodes every time there is new information. In real applications, a sensor on tensile stress may collect data every second, while another sensor on current density collect data every a tenth of a second. The health information of the system can be updated every tenth of a second, without having to performing approximate inference for EM failure mode as often.

Consider a more complex example where the system consists of 50 electrical components that have 2 failure modes. Figure 15 shows a plot of amount of time required as a function of number of failure modes that have the same dependent factor. Assume it takes 1 second to calculate 1,000 iteration of an average marginal distribution computation and an approximate inference requires 10,000 iterations to reach reasonably accurate result. Using the proposed technique, the computation stays roughly the same, while traditional computation time increases exponentially with the number of dependent failure modes.

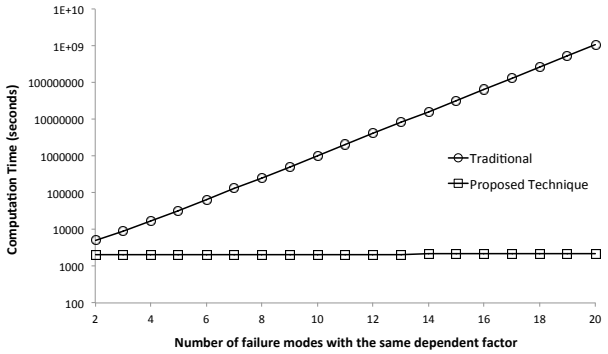


Figure 15: Plot of computation time vs. number of failure modes with the same dependent factor

As mentioned in the previous section, with pre-computation method, the accuracy of inference computation depends mainly on the number of selections of possible values of the variables. Using EM failure mode in the earlier example, Figure 16 shows the average percentage of accuracy against the number of selections of J and T values.

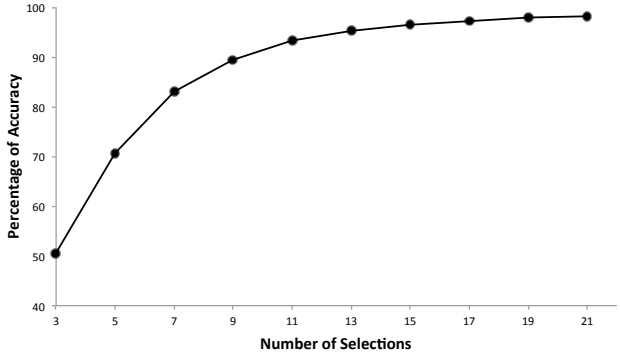


Figure 16: Plot of percentage of accuracy vs. number of selections of J and T values

The optimal number of selections depends on the accuracy required by the particular application and how much pre-computation time is available. In more complex system, the number of selections should also be varied depending on the sensitivity of each variable.

5. CONCLUSION

This research presents new modeling approach, computational algorithms, and an example application for efficient dependency calculation in on-line System Health Management. Hybrid dynamic Bayesian network modeling were introduced with component-based structure and algorithm to represent complex engineering systems in a way that it allows accurate representation of underlying physics of failure by using empirical degradation model with continuous variables. With dynamic hybrid Bayesian Network model requiring Markov Chain Monte Carlo for probabilistic inference, this paper develops computational algorithms that enables monitoring and diagnosing complex systems in real-time. The algorithms use the characteristics of System Health Management applications to allow reduction of number of inference required and reduce the calculation time by the means of pre-computation and dynamic programming.

REFERENCES

Boudali, H., & Dugan, J. B. (2006). A continuous-time Bayesian network reliability modeling, and analysis framework. *Reliability, IEEE Transactions on,* 55 (1), 86-97.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. *Proceedings of the fourteenth conference on uncertainty in artificial intelligence*.

Chen, Z. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics,* 1-69.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence,* 42 (2-3), 393-405.

Cousins, S. B., Chena, W., & Frisse, M. E. (1993). A tutorial introduction to stochastic simulation algorithms for belief networks. *Artificial Intelligence in Medicine,* 5 (4), 315-340.

Dagum, P., & Horvitz, E. (1993). A Bayesian analysis of simulation algorithms for inference in belief networks. *Networks*, 23 (5), 499-516.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence,* 60 (1), 141-154.

Doguc, O., & Ramirez-Marquez, J. E. (2009). A generic method for estimating system reliability using Bayesian networks. *Reliability Engineering & System Safety,* 94 (2), 542-550.

Ferreiro, S., Arnaiz, A., Sierra, B., & Irigoien, I. (2011). A Bayesian network model integrated in a prognostics and health management system for aircraft line maintenance. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering ,* 225 (8), 886-901.

Friedman, N. (1998). The Bayesian structural EM algorithm. In G. F. Cooper (Ed.), *Proceedings of the fourteenth conference on uncertainty in artificial intelligence (UAI-98)* (pp. 129-138). Morgan Kaufmann.

Jensen, F. (2001). *Bayesian Networks and Decision Graph.* Springer.

Langseth, H., & Portinale, L. (2007). Bayesian networks in reliability. *Reliability Engineering & System Safety,* 92 (1), 92-108.

Lauritzen, S. L., & Jensen, F. (2001). Stable local computation with conditional Gaussian distributions. *Stat. & Comp.,* 11, 191-203.

Lerner, U. N. (2002). *Hybrid Bayesian networks for reasoning about complex systems.* Standford University, Dep. of Comp. Sci. Stanford.

Moral, S., Rumi, R., & Salmeron, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks. *ECSQARU 2001. LNCS (LNAI)* (Vol. 2143, pp. 156-167). Springer, Heidelberg.

Murphy, K. (2002). *Dynamic Bayesian networks: representation, inference and learning.* PhD thesis, UC Berkeley, Dept. Computer Science.

Neil, M., Tailor, M., Marquez, D., Fenton, N., & Hear. (2007). Inference in Bayesian networks using dynamic discretisation. *Statistics and Computing,* 17 (3), 219-233.

Pearl, J. (1986). Fusion, propagation and structuring in belief networks. *Artificial Intelligent,* 29, 241-288.

Schumann, J., Rozier, K. Y., Reinbacher, T., Mengshoel, O. J., Mbaya, T., & Ippolito, C. (2013). Towards real-time, on-board, hardware-supported sensor and software health management for unmanned aerial systems. *Annual conference of the prognostics and health managements society.*

Shenoy, P. P. (2006). Inference in hybrid Bayesian networks using mixtures of Gaussians. *Uncertainty in Artificial Intelligence* (pp. 428-436). AUAI Press, Corvallis.

Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., & Tripot, G. (2012). A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models. *Reliability, IEEE Transactions on,* 61 (2), 491-503.

Weber, P., & Jouffe, L. (2006). Complex system reliability modelling with dynamic object oriented Bayesian networks (DOOBN). *Reliability Engineering & System Safety,* 91 (2), 149-162.

Wilson, A. G., & Huzurbazar, A. V. (2007). Bayesian networks for multilevel system reliability. *Reliability Engineering & System Safety,* 92 (10), 1413-1420.

BIOGRAPHIES

Chonlagarn Iamsumang is a Ph.D. candidate in Reliability Engineering at A.J. Clark School of Engineering at the University of Maryland - College Park. He received his M.S. in Nuclear Science and Engineering from MIT in 2010 and B.S. in Physics and Engineering from Brown University in 2006. His current research focuses on applying dynamic

hybrid Bayesian Network for engineering System Health Management.

Ali Mosleh is Distinguished Professor and holder of the Evelyn Knight Chair in Engineering at the University of California in Los Angeles. Prior to that he was the Nicole J. Kim Eminent Professor Chair in Engineering and the Director of the Center for Risk and Reliability at the University of Maryland. He was elected to the US National Academy of Engineering in 2010, and is a Fellow of the Society for Risk Analysis, and the American Nuclear Society, recipient of several scientific achievement awards, and consultant and technical advisor to numerous national and international organizations, including appointment by President George W. Bush to the U.S. Nuclear Waste Technical Review Board, a position in which he continued to serve in the administration of President Obama. He conducts research on methods for probabilistic risk analysis and reliability of complex systems and has made many contributions in diverse fields of theory and application.

Mohammad Modarres is Minta Martin Professor of Engineering and Director of Reliability Engineering at A.J. Clark School of Engineering at the University of Maryland - College Park. His research areas are probabilistic risk assessment and management, uncertainty analysis and physics of failure degradation modeling. He has over 300 papers in archival journals and proceedings of conferences, four books, one handbook, four edited books and five book chapters in various areas of risk and reliability engineering. He is a University of Maryland Distinguished Scholar. He received his PhD in Nuclear Engineering from MIT in 1980 and M.S. in Mechanical Engineering also from MIT.

Prognostic Optimization of Phased Array Antenna for Self-Healing

David Allen[1]

[1]*HRL Laboratories, LLC, Malibu, CA, 90265, USA*
dlallen@hrl.com

ABSTRACT

Phased array antennas are widely used in many applications and consist of many antennas coupled together to enable digital beam-forming. As transmit/receive elements begin to degrade and eventually fail the antenna's beam will distort from the desired pattern. We propose a novel optimization algorithm which takes into account not only the current state-of-health of the system, but potential future states-of-health from prognostic observations. The approach can be run entirely off-line (before the start of a mission), so requires no additional computational resources or sensors be added to the system and does not require the system to be able to detect the degradation/failures during a mission. Our main objective is to trade some current optimization flexibility for improved system robustness under future failures.

1. INTRODUCTION

Phased array antennas (Hansen, 2009) are used in many domains such as radars, communications, satellites, and weather research and many deployed systems exist across airborne, ground, maritime and space domains. They are composed of many individual elements and the radiation pattern depends on each element's location, excitation magnitude and phase. As elements degrade and eventually fail this affects the ability of the array antenna to produce the desired radiation pattern.

Typically the location of the elements is fixed, however by adjusting the excitation magnitude and phase through digital beam-forming the radiation pattern (known as the beam) can be steered, made broader or narrower, regions of enhanced or nulled coverage can be created etc. without any mechanical rotation of the antenna. Array optimization or reconfiguration is the process of generating the parameters for the excitation magnitude and phase of each element to adapt the overall beam to the desired pattern. Most existing approaches are designed for offline use prior to start of a mission or task. They analyze the current state-of-health of the system, such as which elements are fully functional and which are failed, and then performs the optimization. In instances where failures can be detected during the mission these techniques can be rerun to compensate for failures.

The approach presented here assumes that we do not have a way of reliably detecting degradation or failures while in operation, however we may have the ability to detect which elements are at risk of failing in the near future (e.g. maybe they have already begun to degrade or are being heavily stressed by current usage). Additionally some new array materials such as GaN may provide prognostic observables prior to failures. We go beyond current techniques by not only optimizing over the current state-of-health, but also performing a preemptive optimization over potential future states-of-health.

This preemptive optimization is much more robust because it allows us to maintain mission specifications of our system even in the presence of undetected future failures which might occur during the mission. Overall this will help improve the system's affordability and survivability as repairs can be delayed or shifted to more convenient times, such as delaying them till access to external test/repair equipment is available. This graceful degradation or self-healing can lead to important performance improvements.

2. ARRAY OPTIMIZATION

An array's radiation pattern is a function of each element's location, excitation magnitude, and phase. An example beam pattern from a 32 element linear array is depicted in Figure 1. Many techniques exist for determining the excitation magnitude and phase parameters for each element to control the beam.

Some beam control may involve steering the beam or trying to optimize a cost criteria such as maximum side-lobe level (SLL), average SLL, or cumulative difference. Many techniques have been developed over the years to optimize the desired beam pattern. Some of the most common include genetic algorithms (Yeo & Lu, 1999), stochastic optimization (such as Particle Swarm Optimization - PSO)

David Allen. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

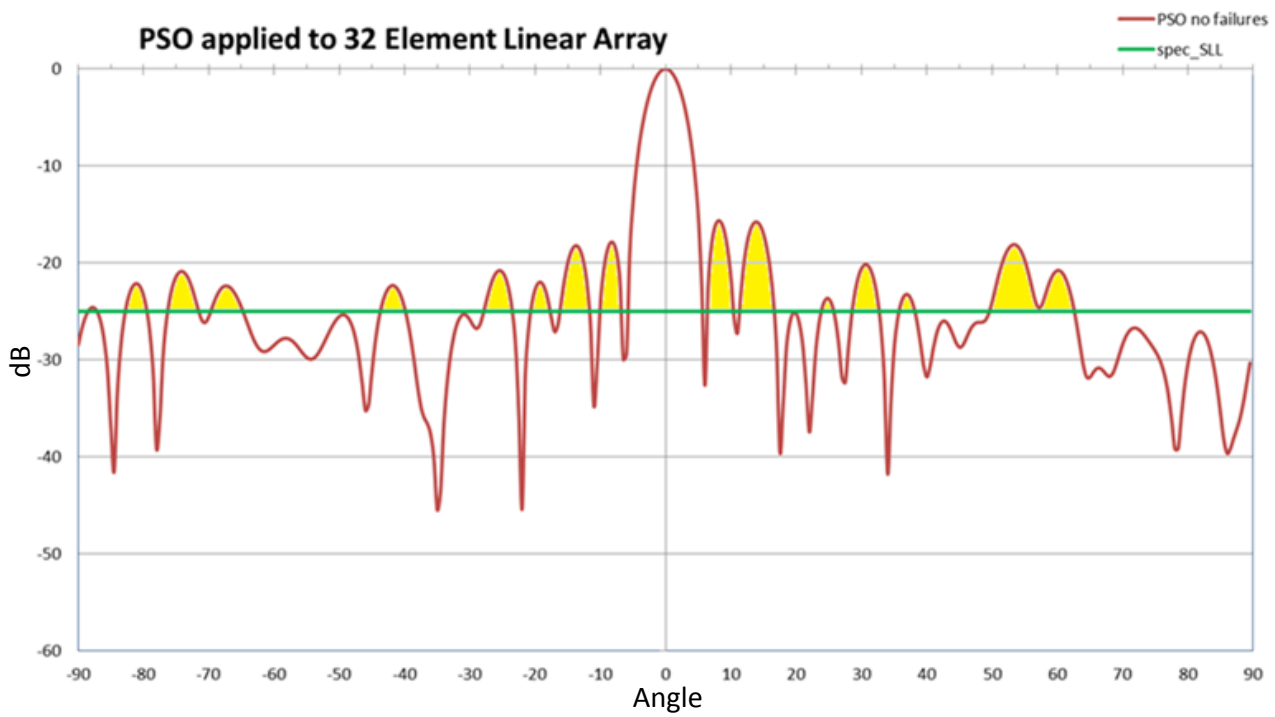


Figure 1. Radiation pattern generated from a 32 element linear array.

(Yeo & Lu, 2009)(Boeringer & Werner, 2004)(Khodier & Al-Aqeel, 2009), and hybrid approaches (Yeo & Lu, 2005).

Some approaches have also been developed to handle re-optimization after element failures (Joler, 2012)(Keizer, 2007). These are mostly performed off-line prior to a mission. There have been some attempts at detecting failures while the array is in use and doing very efficient heuristic compensation (Levitas et al., 1999).

The radiation pattern can be generated from the array factor (AF) given by (Boeringer & Werner, 2004):

$$AF(\Theta) = \sum_{n=1}^{N} A_n e^{j2\Pi n(d/\lambda)\sin(\Theta)} \quad (1)$$

where N is the number of radiating elements, A_n are the complex element weights for excitation magnitude and phase, and $d/\lambda = ½$ is the spacing between elements normalized by the wavelength.

Particle Swarm Optimization

Particle swarm optimization (PSO) is a generic optimization approach to iteratively improve the current best solution with regard to a given metric and has been used extensively for optimizing phased array antennas. The basic concept is that there is a swarm of particles where each is a possible solution (i.e. a setting of all elements' excitation magnitude and phase parameters). These particles move through the solution space based on their own local observations and also the best known position of the swarm in the overall search-space. This allows it to be guided to regions of known good quality while still allowing particles to explore unknown regions in search of better solutions. In practice, as the algorithm progresses the particles will move toward near-optimal solutions.

Algorithm 1 presents the pseudo code for PSO. After initialization it iteratively updates each particle's velocity and position; then it computes a cost function to determine if the position is better than previously observed positions. PSO is therefore general enough that it can optimize over various different cost functions.

The results of PSO are not guaranteed to be optimal, however in practice the optimization converges to near optimal results fairly quickly and in many instances have been shown to outperform other approaches such as genetic algorithms (Yeo & Lu, 2009)(Boeringer & Werner, 2004).

Some typical cost functions used for phased array optimization include:

Maximum side-lobe level (or Peak side-lobe level): the largest side peak of the beam pattern relative to the main beam

Average side-lobe level: the average of the side-lobe peaks relative to the main beam

Cumulative Difference: the area under the beam pattern but above a specified threshold (ignoring the main beam) (see portion shaded yellow in Figure 1).

Algorithm 1. Particle Swarm Optimization (PSO) Pseudocode

1. Initialization
   a. For all particles
      i. Set position uniformly distributed $x_i = U(b_{low}, b_{up})$;
         where b defines the search space and low is the lower bound and up is the upper bound
      ii. Set velocity $v_i=U(-|b_{up}-b_{low}|, |b_{up}-b_{low}|)$
      iii. Initialize the particle's best known position (p_i)
   b. Initialize swarm's best known position (g)
2. Repeat until termination criteria met
   a. For each particle i
      i. Update the velocity (v_i,d) for each dimension d based on PSO update function
      ii. Update the position $x_i = x_i+v_i$
      iii. Compute cost function: $f(x_i)$
      iv. If($f(x_i)$ better than $f(p_i)$)
         1. $p_i = x_i$; Update the particle's best known position
      v. If($f(x_i)$ better than f(g))
         1. $g = x_i$; Update the swarm's best known position
3. Return g, the best solution found

3. PREEMPTIVE OPTIMIZATION ALGORITHM

As elements of the phased array antenna fail the radiation pattern will get distorted. For example the main beam may broaden out or the side-lobes may increase above the desired threshold. If you can detect the failure while the array is in the field then you can re-optimize the pattern to compensate for the distortion or degradation (Keizer, 2007). In many systems the engineering cost to add additional sensors to reliably detect the failures is prohibitive and therefore failures cannot be detected while system is in use and external test equipment unavailable. However in some instances we may be able to detect potential future failures, such as elements that have not completely failed but have partially degraded, or elements which have been heavily stressed in the past, or those where we have prognostic observations predicting failure onset.

In this work we propose a new optimization approach which not only leverages the system's current state-of-health, but its potential future states. Current algorithms monitor the current state-of-health and assume it is fixed, however in the real-world those elements will begin to degrade and eventually fail. If left uncorrected, these can significantly affect the performance of the array. Our novel optimization approach works by adapting the cost function used by the PSO algorithm.

For simplicity we will assume either an element is failed or not (the algorithm can be extended to handle the case of degradation).

Let F be a list of currently failed elements, (e.g. $F = \{3,4,6,7\}$).

Let P be a list of potential future failures, (e.g. $P = \{5,20,30\}$).

The standard approach to optimization would compute PSO(F). It takes the current state-of-health as input and a previously defined cost metric. The optimization generates a set of element parameters optimizing the beam with respect to the cost metric. Failures of elements in F can be modeled by setting the excitation magnitude of those elements to 0.

Our approach, PSO_Robust(F, P), takes as input both the current state-of-health and a list of potential future element failures. In Algorithm 1, on line (2.a.iii) a cost function $f(x_i)$ is computed. The input to this function is a current instantiation of all the elements' parameters (hence with it you can compute the radiation pattern such as in Figure 1 and compute the cost functions previously described).

We will replace the cost function $f(x_i)$ with the following:

$$f_robust(x_i, P) = \sqrt[|P|+1]{f(x_i) * \prod_{p \in P} f(x_i, x_p = 0)} \quad (2)$$

where |P| is the cardinality of P.

What the above cost function does is compute the cost under the current-state-of-health, $f(x_i)$, and under each potential future state $f(x_i, x_p=0)$ under the assumption of single future failure. This function then combines these results using the geometric mean. Other approaches could be used to combine the results (e.g. arithmetic mean [average] of the costs, weighted combination of current and average of potential states, etc.). We chose to use geometric mean because it more heavily weights bad instances than the others. For example under the above scenario where our potential future failures are 5, 20, and 30, if all degrade evenly it would not matter which cost function we chose, but let us assume a failure at 20 or a failure at 30 would degrade performance by a small amount but a failure at 5 would severely impact performance since we already have 3,4,6&7 failed and losing 5 creates a large clustered failure (i.e. no radiation from five consecutive array elements). If x_n handles the future case of a failure at 5 better than x_m then we would like the cost function to measure that, and in general we are more worried about worst case single failures (as they may potentially happen) more than average case results. This ensures that if that failure happens we will maintain our mission specifications under that specific condition rather than the average of all future states.

Table 1. Results showing the max peak SLL and cumulative difference cost function for standard PSO (top) and our robust extension (bottom) under the two cases of no failures and element 5 failing.

PSO	Max Peak SLL	Cumulative Difference
No Failures	-16.47	49.63
Failures: #5	-14.16	69.53

Robust {5}	Max Peak SLL	Cumulative Difference
No Failures	-16.46	59.92
Failures: #5	-16.95	40.86

(SLL threshold = -20dB, swarm=3000, epochs=100)

This approach does incur a penalty for this improved robustness. The generated radiation pattern under the current state-of-health will not be quite as good with respect to the cost metric, however if any of the potential failures occurs it will maintain a more desirable beam pattern. Additionally the metrics can be analyzed a priori to determine if failures would result in performance below mission specifications. This improved robustness, at the expense of a reduced performance, can be very desirable in many application domains such as where online detection of failures is infeasible either technically or due to cost of additional sensors.

4. EXPERIMENTAL RESULTS

We have implemented the above algorithm and performed experiments on a linear array, however the approach is completely general and could be used for other array configurations such as two dimensional arrays. The results we present are using the cumulative difference cost function, but we have experimented with other various cost functions with similar results.

Table 1 shows results where the current health of the system is fully functional and the only potential future failure is element 5. Running PSO results in a beam pattern where the cumulative difference is 49.63, whereas the robust version's difference is 59.92 (lower is better). These correspond with the Max Peak SLL shown on the left, where the difference is only .01 dB (-16.46 vs. -16.47). However if element 5 does fail the cost metric for the PSO optimized beam shoots up to 69.53 compared with only 40.86 for the robust version. Similarly this results in a peak side-lobe level which is 2.5 dB better.

Figure **2** depicts the beam patterns of both the standard PSO and our robust extension under the case of a fully health array and Figure **3** depicts them in the case of an undetected or uncompensated failure at element 5. As can be seen in Figure **2** both algorithms have relatively similar main beams and peak side-lobe levels, however under future failure of element 5 (Figure **3**) the side-lobe closest to the main lobe jumps dramatically under standard PSO, while the robust PSO can still maintain similar performance.

For a slightly more complex case, we look at Table 2, where the current state-of-health has elements 3, 4, 6, & 7 all failed and 5, 20, & 30 are potential future failures. In this case our initial penalty for incorporating robustness is only 5.39 (34.57 vs. 29.19), but under any failure the benefit is fairly substantial (119.72, 118.11, and 63.19). Figures 4-7 show the patterns for the current state-of-health of the array as well as for each of the 3 potential states of the array. Similar to the previous example under the different future failures the robust algorithm does in face maintain not only a better cumulative cost function (which is what it optimized over) but the peak side-lobes also are maintained. If we directly optimize peak SLL we might see even further improvements, however our goal was to maintain the entire pattern, hence the choice of the cumulative difference metric.

Table 2. Results showing the cumulative cost function from the array pattern computed under the failed elements 3, 4, 6, & 7. The first row shows the penalty paid by incorporating the robustness, but in the instances when either 5, 20, or 30 failed there is a substantial benefit.

Failures	PSO	Robust[3,4,6,7]+{5,20,30}	Difference
3, 4, 6, 7	29.19	34.57	5.39
3, 4, 6, 7, 5	262.04	142.32	-119.72
3, 4, 6, 7, 20	352.31	234.20	-118.11
3, 4, 6, 7, 30	165.95	102.76	-63.19

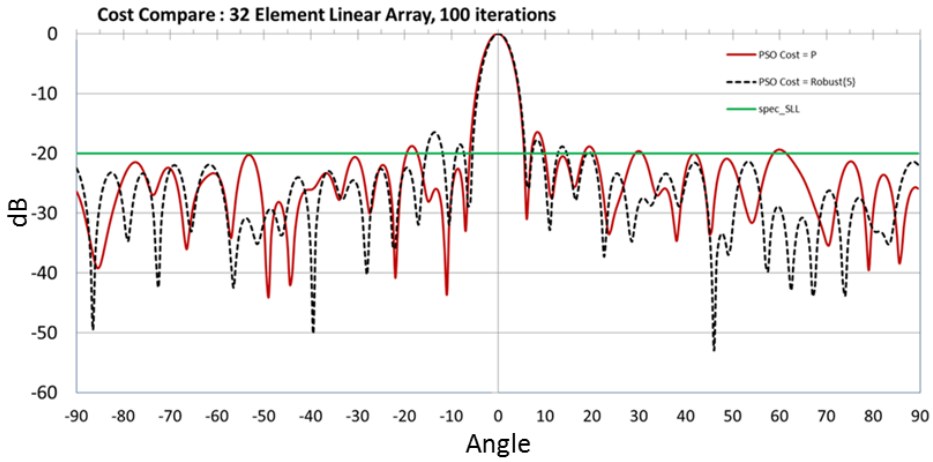


Figure 2. Beam pattern for standard PSO (red) and robust PSO (black-dashed), where there were originally no failed elements and the only potential failure used by the robust version was element 5.

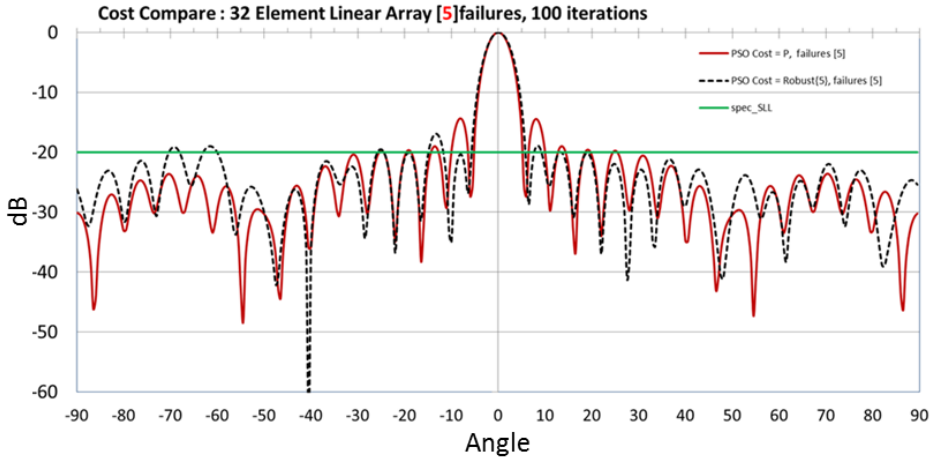


Figure 3. Beam pattern for standard PSO (red) and robust PSO (black-dashed), where they were optimized with no failed elements, but then element 5 did fail. Our robust extension to PSO was able to maintain lower peak side-lobe levels than standard PSO.

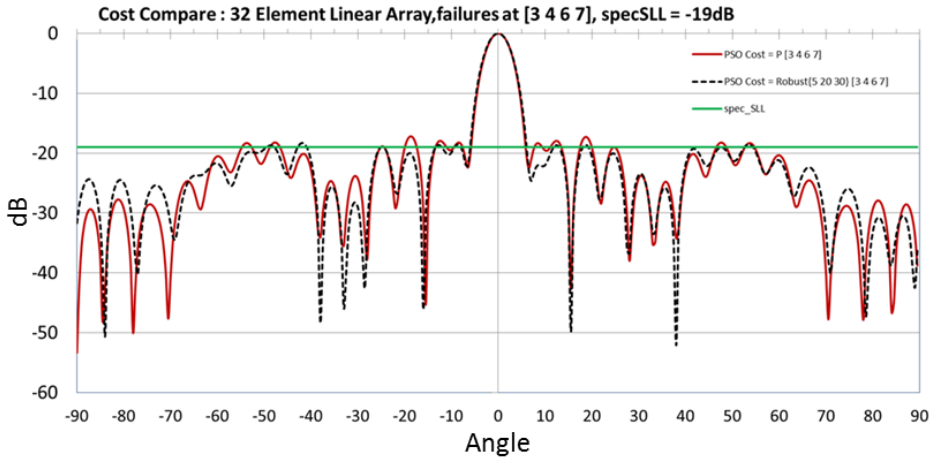


Figure 4. Beam pattern for standard PSO (red) and robust PSO (black-dashed) where elements 3, 4, 6, and 7 were originally failed and elements 5, 20, and 30 were potential future failures.

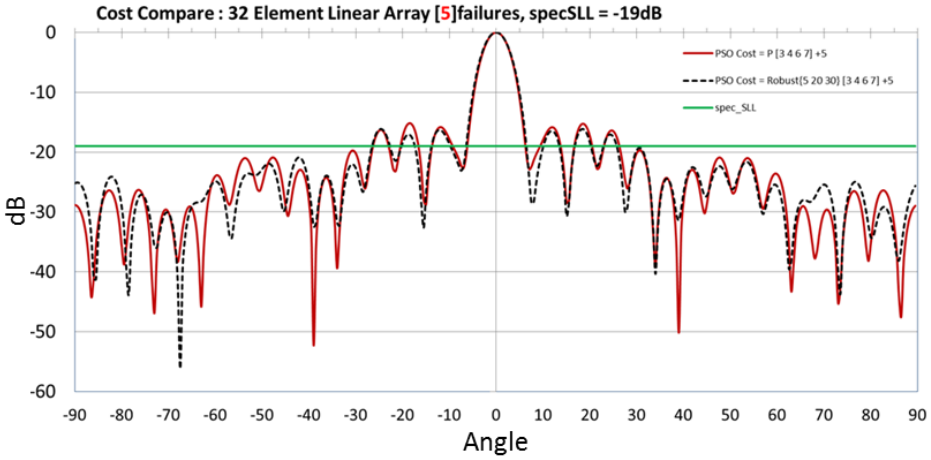


Figure 5. Beam pattern for standard PSO (red) and robust PSO (black-dashed) where elements 3, 4, 6, and 7 were originally failed prior to the optimization and element 5 later failed.

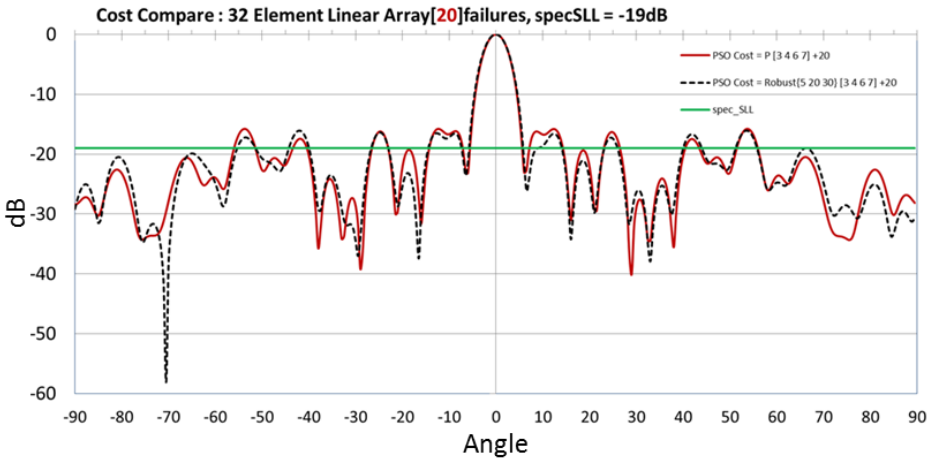


Figure 6. Beam pattern for standard PSO (red) and robust PSO (black-dashed) where elements 3, 4, 6, and 7 were originally failed prior to the optimization and element 20 later failed.

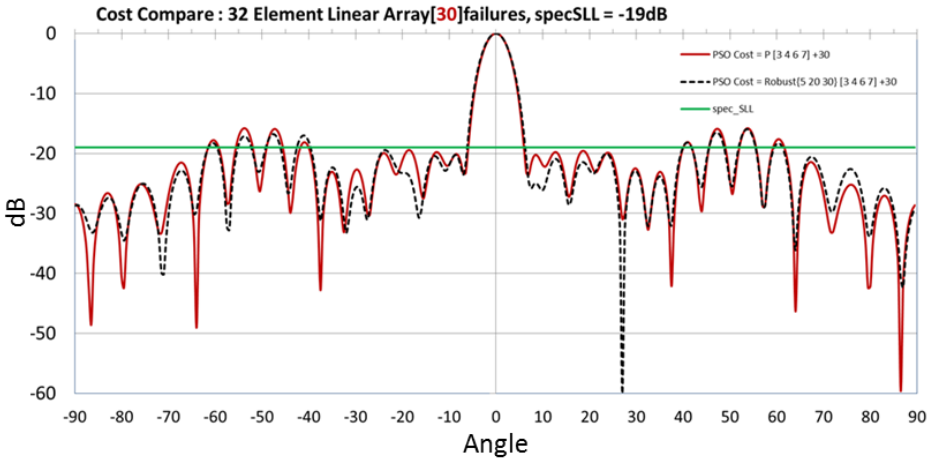


Figure 7. Beam pattern for standard PSO (red) and robust PSO (black-dashed) where elements 3, 4, 6, and 7 were originally failed prior to the optimization and element 30 later failed.

5. CONCLUSIONS

In this work we propose a novel prognostic approach to do preemptive optimization of phased array antennas. When determining element parameters for excitation magnitude and phase during digital beam-forming we not only optimize over the current state-of-health but consider potential future states-of-health. This allows the algorithm to trade some current optimization flexibility for improved system robustness under future failures which might occur during a mission. This improves the overall system's affordability and survivability as it is more robust to failures and repairs can be performed at more optimal times. This technique does assume that potential future failures can be determined, however there is evidence that in many systems this is true. Additionally this approach does not require additional sensors or engineering to reliably detect failures during a mission and does not require systems resources while online, as it is performed prior to the start of a mission but then has the most effect when failures do occur. It also allows a user to determine whether the system will be able to maintain minimum mission specifications even under potential failures a priori, allowing them to make a decision whether to go ahead with the mission.

ACKNOWLEDGEMENT

The author would like to thank Boeing for their support of this project and the technical discussions and implementation work of Gavin Holland, Edward Sabatka, and Marian Kis.

REFERENCES

Boeringer, D.W., & Werner, D.H., (2004). Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis. *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 3.

Hansen, R.C., (2009). *Phased Array Antennas*, Second Edition, John Wiley & Sons, Inc.

Joler, M., (2012). Self-recoverable antenna arrays. *IET Microwaves, Antennas & Propagation*.

Keizer, W., (2007). Element Failure Correction for a Large Monopulse Phased Array Antenna with Active Amplitude Weighting. *IEEE Transactions on Antennas and Propagation*, Vol. 55, No. 8.

Khodier, M., & Al-Aqeel, M., (2009). Linear and Circular Array Optimization: A Study Using Particle Swarm Intelligence. *Progress in Electromagnetics Research B*, Vol. 15.

Levitas, M., Horton, D.A., & Cheston, T.C., (1999). Practical Failure Compensation in Active Phased Arrays. *IEEE Transactions on Antennas and Propagation*, Vol. 47, No. 3.

Yeo, B-K., & Lu, Y., (1999). Array Failure Correction with a Genetic Algorithm. *IEEE Transactions on Antennas and Propagation*, Vol. 47, No. 5.

Yeo, B-K., & Lu, Y., (2005). Adaptive array digital beamforming using complex-coded particle swarm optimization-genetic algorithm. *Microwave Conference*.

Yeo, B-K., & Lu, Y., (2009). Fast array failure correction using improved particle swarm optimization. *Microwave Conference*.

BIOGRAPHIES

David Allen received his M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles (UCLA) in 2001 and 2005 respectively. He joined HRL Laboratories, LLC, in 2006 and currently holds the position of Research Staff Scientist. His research interests include artificial intelligence, probabilistic reasoning, data science, decision making under uncertainty, complex systems analysis, and network science; he has applied them to many domains including integrated system health management, cybersecurity, social network analysis, and behavior modeling. In 2010 he received HRL's Distinguished Inventor Award and in 2011 received a GM Contribution Award.

Multiple-imputation-particle-filtering scheme for Uncertainty Characterization in Battery State-of-Charge Estimation Problems with Missing Measurement Data

David E. Acuña[1], Marcos E. Orchard[1], Jorge F. Silva[1], and Aramis Pérez[1]

[1] *Universidad de Chile, Department of Electrical Engineering. Av. Tupper 2007, Santiago, Chile*
davacuna@ing.uchile.cl
morchard@ing.uchile.cl
josilva@ing.uchile.cl
aramis.perez@ing.uchile.cl

ABSTRACT

The design of particle-filtering-based algorithms for estimation often has to deal with the problem of missing observations. This requires the implementation of an appropriate methodology for real-time uncertainty characterization, within the estimation process, incorporating knowledge from other available sources of information. This article presents preliminary results of a multiple imputation strategy used to improve the performance of a particle-filtering-based state-of-charge (SOC) estimator for lithium-ion (Li-Ion) battery cells. The proposed uncertainty characterization scheme is tested and validated in a case study where the state-space model requires both voltage and discharge current measurements to estimate the SOC. A sudden disconnection of the battery's voltage sensor is assumed to cause significant loss of data. The results show that the multiple-imputation particle filter enables reasonable uncertainty characterization for the state estimate as long as the voltage sensor disconnection continues. Furthermore, when the voltage measurements are once more available, the level of uncertainty adjusts to levels that are comparable to the case where data was not lost.

1. INTRODUCTION

During the last century there has been an increase in production and development of electronics that has changed the way of living. Due to the increasing scarcity of oil, an imminent migration to alternative kinds of energy becomes relevant. The automotive industry has been putting research efforts into the development of energy storage devices (ESDs) for the production of hybrid electric vehicles (HEV) or fully electric vehicles (EV). As a result, ESDs have been playing a crucial role regarding autonomy of systems. This last fact has impulsed research on Li-Ion battery cells due to advantages over other types of ESDs, being its larger charge density by unit of mass or volume one of the most important features. From the automotive industry, the concept of *"Battery Management Systems"* (BMS) (Pattipati, Sankavaram, & Pattipati, 2011) rises naturally looking for systems capable of providing protection and optimal operating conditions for batteries while accounting for life predictions through monitoring acquired data and interfacing external modules. Regarding this, the *"State-of-Charge"* (SOC) (Pattipati et al., 2011) -quantifing the remaining available energy stored-, the *"State-of-Health"* (SOH) (Pattipati et al., 2011) -describing the degree of degradation-, and the *"Remaining Useful Life"* (RUL) (Orchard & Vachtsevanos, 2009) generate important information about the actual battery cells for optimal management. Unfortunately, due to incapability to measure them directly in an online framework, BMS systems must incorporate real-time estimation and prediction routines to carry out their objectives.

These routines heavily depend on real-time measurements for their implementation, and when measuring data from any device it is possible to miss information due to, for example, transmission problems within sensor networks. Completing the acquired data set is not just as simple as filling in the missing information with averaged values. In this regard, many strategies may be adopted to solve the problem of sequential state estimation with incomplete data sets. Among them, single imputation methods fail due to the lack of uncertainty characterization. In (Rubin, 1987) the idea of multiple imputations was proposed. This method considers different values for each missing datum and combines their induced probability distributions into a single solution for parameter estimation. This led to the multiple imputation particle filter (Housfater, Zhang, & Zhou, 2006), where particle-

David E. Acuña et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

filtering methods (Andrieu, Doucet, & Punskaya, 2001) were used, taking into account uncertainty of missing data through a multiple imputation strategy.

This work presents an improvement of the particle-filtering-based Bayesian approach adopted by (Orchard, Cerda, Olivares, & Silva, 2012) for real-time uncertainty characterization in SOC estimation for Li-Ion batteries, based on a multiple-imputation strategy. The validation case for this proposed Multiple Imputation Particle Filter algorithm considers a situation where 1000 sequential voltage measurements are assumed to be lost, emulating the disconnection of the associated sensor during the execution of a specific discharge cycle. Obtained results show that the uncertainty associated to the state estimate due to lost data is bounded. Furthermore, those uncertainty bounds are smaller than those obtained when simply discarding incomplete measurements and applying n-step prediction to generate the prior state density function.

The article is structured as follows. In Section 2, a theoretical background is presented reviewing the underlying concepts of particle filters and the multiple imputation strategy. In Section 3, a new multiple-imputation-based particle filter is applied in Li-Ion battery cells for SOC estimation when voltage and discharge current are measured. Sudden disconnections of the battery's voltage sensor are simulated and uncertainty characterization is analyzed. Finally, conclusions and future work are presented in Section 4.

2. THEORETICAL BACKGROUND

Real world systems are commonly dynamic, nonlinear, and may involve a high dimensionality relationship between variables. In this regard, state-space models offer a good treatment for these systems; for example, when monitoring critical system components which physical phenomenology may be modeled directly under the state-space form. Moreover, uncertainty due to noisy measurements associated with sensors constrains or other sources of disturbances such as the lack of knowledge about the actual system dynamics, can be incorporated into the state-space form with ease. This allows to adopt a Bayesian approach, where the main objective is to estimate the underlying probability distribution in order to perform statistical inferences. Since the analytical solutions may be founded under certain conditions, the real problem to be addressed is that of evaluating complex integrals where numerical methods tend to breakdown, even more when high dimensional systems are involved. An alternative to address this problem is the use of particle filters, which is presented in the following section. Later, an introduction to multiple imputation for dealing with missing data and the way multiple imputation particle filter is presented.

2.1. Particle Filters

Due to the employment of digital computers for signal processing, it is of interest to develop a Bayesian processor where measurements arrive sequentially in time. The recursive estimation of the evolving posterior distribution is the so called *optimal filtering* problem. A mathematical framework is provided below for solving this problem using particle filters.

Let $X = \{X_t, t \in \mathbb{N}\}$ be a first order Markov process denoting a n_x-dimensional system state vector with initial distribution $p(x_0)$ and transition probability $p(x_t|x_{t-1})$. Also, let $Y = \{Y_t, t \in \mathbb{N} \setminus \{0\}\}$ denote n_y-dimensional conditionally independent noisy observations. The whole system is represented in state-space form as

$$x_t = f(x_{t-1}, w_{t-1}) \quad (1)$$

$$y_t = g(x_t, v_t) \quad (2)$$

where w_t and v_t denote independent random variables whose distributions are not necessarily Gaussian. Since it is difficult to compute the filtering posterior distribution $p(x_t|y_{1:t})$ directly, Bayesian estimators are constructed from *Bayes' rule*.

Under Markovian assumptions, the filtering posterior distribution can be decomposed into

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t) \cdot p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \quad (3)$$

In this context, sequential Monte Carlo methods (SMC) offer an alternative to numerical integration techniques that fail due to high computation. SMC methods, also called particle filters, are stochastic computational techniques designed for simulating highly complex systems in an efficient way. In Bayesian estimation, these techniques simulate probability distributions by using a collection of N weighted samples or *particles*, $\{x_t^{(i)}, \mathcal{W}_t^{(i)}\}_{i=1}^N$, that yields to discrete mass probability distributions, as shown in Eq (4).

$$\hat{p}(x_t|y_{1:t}) \approx \sum_{i=1}^{N} \mathcal{W}_t^{(i)} \delta(x_t - x_t^{(i)}) \quad (4)$$

The weighting process is made by applying the *sequential importance resampling* (SIR) algorithm, which is explained in the following subsections.

2.1.1. Sequential Importance Sampling

The concept of importance sampling is used to simulate samples from a proposed distribution in order to estimate a posterior distribution. The key point for a successful sampling is to choose appropriately the importance distribution. Sampling from posterior distributions is a common task in order to get Monte Carlo (MC) estimates. However, it is not feasible most of the time since it becomes computationally intensive.

For example, Eq (5) shows the calculation of expectations.

$$\hat{f}(x_t) = E_{X|Y}\{f(x_t)\} = \int_X f(x_t)p(x_t|y_{1:t})dx_t \quad (5)$$

Drawing N independent identical distributed random samples from $p(x_t|y_{1:t})$, the integral may be approximated by a sum of delta-Dirac functions.

$$\hat{f}(x_t) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_t)\delta(x_t - x_t^{(i)}) \quad (6)$$

$$= \frac{1}{N}\sum_{i=1}^{N} f(x_t^{(i)}) \quad (7)$$

These approximations may not hold when it is not possible to sample directly from $p(x_t|y_{1:t})$, thus the *sequetial importance sampling* (SIS) algorithm avoids these difficulties by drawing samples from an *importance distribution* approximating the targeted posterior distribution by appropriate weighting. The weights are recursively defined as

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t|\tilde{x}_t^{(i)}) \cdot p(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{\pi(\tilde{x}_t^{(i)}|\tilde{x}_{0:t-1}^{(i)}, y_{1:t})} \quad (8)$$

where $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ is a set of N random samples drawn from the importance distribution $\pi(\tilde{x}_t^{(i)}|\tilde{x}_{0:t-1}^{(i)}, y_{1:t})$. Also, defining normalized weights

$$\mathcal{W}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}} \quad (9)$$

then the posterior distribution can be approximated by the expression described in Eq (4).

2.1.2. Resampling

When the updating process begins, a tendency to increase the variance of particles is seen, setting negligible weights to some of them. These particles become useless as they track low probability paths of the state vector. In order to solve this problem, a *resampling* step is incorporated, which leads to the SIR algorithm.

An analytical expression for measuring how degenerated are the particles is given by the *effective particle sample size* showed in Eq (10).

$$N_{eff}(t) = \frac{N}{1 + Var_{p(\cdot|y_{1:t})}(w(x_t))} \quad (10)$$

As it is not possible to calculate N_{eff}, an estimate is given by

$$\hat{N}_{eff}(t) = \frac{1}{\sum_{i=1}^N (\mathcal{W}_t^{(i)})^2} \quad (11)$$

In other words, the resampling step consist of removing small weighted particles while retaining and replicating those of large weights. Thus, whenever $\hat{N}_{eff} \leq N_{thres}$, with N_{thres} a fixed threshold, the depletion of the particles is imminent and resampling must be applied.

Algorithm 1 SIR Particle Filter

1. Importance Sampling
   for $i = 1, \ldots, N$ **do**
   - Sample $\tilde{x}_t^{(i)} \sim \pi(x_t|x_{0:t-1}^{(i)}, y_{1:t})$ and set $\tilde{x}_{0:t}^{(i)} \triangleq (x_{0:t}^{(i)}, \tilde{x}_t^{(i)})$
   - Compute the importance weights $w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t|\tilde{x}_t^{(i)}) \cdot p(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{\pi(\tilde{x}_t^{(i)}|\tilde{x}_{0:t-1}^{(i)}, y_{1:t})}$
   - Normalize $\mathcal{W}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}$

   end for

2. Resampling
   if $\hat{N}_{eff} \geq N_{thres}$ **then**
   for $i = 1, \ldots, N$ **do**
   - $x_{0:t}^{(i)} = \tilde{x}_{0:t}^{(i)}$

   end for
   else
   for $i = 1, \ldots, N$ **do**
   - Sample an index $j(i)$ distributed according to the discrete distribution satisfying $P(j(i) = l) = \mathcal{W}_t^{(i)}$ for $l = 1, \ldots, N$
   - $x_{0:t}^{(i)} = \tilde{x}_{0:t}^{j(i)}$ and $w_t^{(i)} = \frac{1}{N}$

   end for
   end if

In general, the SIR particle filter is divided into two steps. Firstly, a *prediction* is done using the state transition model to generate the prior distribution $p(x_k|x_{k-1})$. Then an *update* step is done to modify the particle weights through the likelihood $p(y_k|x_k)$. If the resulting particles are degenerated, a $resampling$ step is added, as it was shown previously.

2.2. Multiple imputations

Missing data is a problem that may be treated mainly from two perspectives. On the one hand, *single imputation* techniques fill the incomplete data set imputing single values at each missing datum. The advantage of this perspective is that it allows standard complete data methods to be used. However, these techniques fail due to the lack of uncertainty characterization of both, the sampling variability and the uncertainty associated with the imputation model. On the other hand, the idea of *multiple imputations* retains the advantages of single imputation techniques and also accounts for the uncertainty of the missing mechanism. Multiple imputations (Rubin, 1987) consist of creating multiple complete data sets imputing m values for each missing datum so that sampling variability around the actual values is incorporated for performing valid inferences. Nevertheless, multiple imputations

has disadvantages like the need of drawing more imputations and larger memory space for storing and processing multiple-imputed data sets.

An important issue is the task of choosing the right number of imputations (Graham, Olchowski, & Gilreath, 2007). Obviously, the computational cost is higher as the number of imputations increases. In this regard, (Rubin, 1987, p. 114) shows that an approximation of efficiency for an estimate is given by

$$(1 + \frac{\gamma}{m})^{-1/2} \tag{12}$$

in units of standard errors, where m is the number of imputations and γ is the fraction of missing information in the estimation. Consequently, excellent results may be obtained using only few imputations ($m = 3, 4, 5$).

2.3. Multiple Imputation Particle Filter

Originally introduced by (Housfater et al., 2006), the Multiple Imputation Particle Filter extends the PF algorithm by incorporating a multiple imputation (MI) procedure for cases where measurement data is not available, so that the algorithm can include the corresponding uncertainty into the estimation process. The main statistical assumption in this approach is that the missing mechanism is *Missing at Random (MAR)*, thus, it does not depend on the missing measures given the observed ones.

For readability, a change in notation is necessary. As it was stated in (Housfater et al., 2006), lets denote now the measurements as a partitioned vector $U_t = (Z_t, Y_t)$, where Z_t corresponds to the missing part and Y_t is from now on the observed part. Then, the MI PF algorithm performs the same as the SIR PF except that there are missing measures. In this case, a MI strategy is adopted.

An imputation model expressed as a probability distribution ϕ is required for drawing m samples -imputations-, that is

$$z_t^j \sim \phi(z_t|y_{1:t}) \tag{13}$$

where $j = \{1, \ldots, m\}$ denotes the imputation index. Similarly to importance sampling, each imputation is associated with a weight p_t^j holding the condition $\sum_{j=1}^{m} p_t^j = 1$. Acording to (Liu, Kong, & Wong, 1994), the filtering posterior distribution may be expressed as

$$p(x_t|y_{1:t}) = \int p(x_t|u_{1:t-1}, y_t)p(z_t|y_{1:t})dz_t. \tag{14}$$

By performing a Monte Carlo approximation yields

$$p(x_t|y_{1:t}) \simeq \sum_{j=1}^{m} p_t^j p(x_t|u_{1:t-1}, u_t^j), \tag{15}$$

where $u_t^j = (z_t^j, y_t)$ are complete data sets formed from imputed values. Additionally, by particle filtering each of these data sets yields

$$p(x_t|u_{1:t-1}, u_t^j) \approx \sum_{i=1}^{N} w_t^{(i,j)} \delta(x_t - x_t^{(i,j)}), \tag{16}$$

where the indexes i and j indicate the particle and the imputation, respectively. Thus, an approximation of the desired posterior distribution is

$$p(x_t|y_{1:t}) \approx \sum_{j=1}^{m} \sum_{i=1}^{N} p_t^j w_t^{(i,j)} \delta(x_t - x_t^{(i,j)}). \tag{17}$$

3. MULTIPLE-IMPUTATION-BASED UNCERTAINTY CHARACTERIZATION FOR SOC ESTIMATION

The SOC is conceived as a quantification of the available energy stored regarding the actual rated capacity, but as a percentage. It conforms an important feature to address for systems' autonomy when they are energized by ESDs, either as main sources or as a backup. As it is not possible to directly measure the SOC, estimation and prognosis algorithms must be addressed for getting valid predictions from usually noisy measurements like current, voltage and temperature, while carrying out a proper management of the system. Actually, knowledge about it is essential for control of autonomous systems where the End-of-Discharge (EoD) time plays a key role.

According to (Orchard et al., 2012), a wide variety of methods have been proposed in the literature for modeling batteries in offline applications; e.g., electrochemical models. Other methods, more suitable for online implementations, are based on open-circuit voltage (OCV) representations. These methods relate directly the SOC and measured voltage but requires large resting periods for batteries, being inefficient for online estimation. The *"Electrochemical Impedance Spectroscopy"* (EIS) method requires costly equipment, being infeasible for practical applications. In this regard, research efforts have focused on developing estimation and prognosis algorithms based on phenomenological relations through fuzzy logic, neural networks and Bayesian frameworks (Orchard et al., 2012), among others. The main problem in all these cases is that these approaches assume complete data sets for state/parameter estimation purposes.

3.1. State-Space Model for Lithium-Ion Batteries

One of the main advantages of adopting a particle-filtering approach for estimation under noisy measurement data is that prior knowledge about the systems dynamics can be directly incorporated into the model as well as its associated uncertainties. Also, it is possible to capture critical physical phenomenology directly into the state-space form, relating it to an observation model which enables the convergence to the

true estimates through the likelihood of sequential measurement data.

Proposed by (Pola, 2014), the state-space model for lithium-ion battery cells used is the following.

State transition model

$$x_1(t+1) = x_1(t) + w_1(t) \quad (18)$$

$$x_2(t+1) = x_2(t) - \frac{v(t) \cdot i(t) \cdot \Delta t}{E_{crit}} + w_2(t) \quad (19)$$

Measurement equation

$$v(t) = v_L + (v_0 - v_L) \cdot e^{\gamma \cdot (x_2(t)-1)} + \alpha \cdot v_L \cdot \ldots \\ \ldots (x_2(t) - 1) + (1 - \alpha) \cdot v_L \cdot (e^{-\beta} - \ldots \\ \ldots e^{-\beta \cdot \sqrt{x_2(t)}}) - i(t) \cdot x_1(t) + \eta(t) \quad (20)$$

where $w_1(t) \sim \mathcal{N}(0, \sigma_1)$ and $w_2(t) \sim \mathcal{N}(0, \sigma_2)$ correspond to additive white Gaussian noise and $\eta(t) \sim \mathcal{N}(0, \sigma_{obs})$ is also a normal distributed random variable accounting for measurement uncertainties. The sample time $\Delta t[sec]$ and the current $i(t)[A]$ are considered input variables whereas the battery voltage $v(t)[V]$ is considered the system's output. The state variables $x_1(t)$ and $x_2(t)$ are chosen strategically under physical meaning as the internal resistance and the SOC, respectively. Finally, as the SOC is expressed as a percentage of energy, E_{crit} represents a normalizing constant whose units are $[V\,A\,sec]$. All other model parameters are assumed to be known constants within each battery discharge cycle. Their values are obtained by following the procedure described in (Pola, 2014), and applying it to data that should be obtained from a complete discharge cycle at constant (nominal) discharge current.

3.2. Implementation of a Multiple Imputation Strategy

(Orchard et al., 2012) proposed a detailed procedure for estimation and prognosis for the SOC. However, what happens when sudden disconnections (or data losses) affect sensors' performance? Perhaps, SOC estimates may be eventually biased, affecting deeply the whole estimation stage and providing invalid information, and the system's autonomy would no longer be guaranteed. In this regard, a new approach from the Multiple Imputation Theory is proposed for uncertainty characterization in particle-filtering-based SOC estimators where voltage measurements are missing during extended periods of time (while discharge current measurements are always available). Future work will focus on the case when battery discharge current measurements are lost instead.

The Multiple Imputation Particle Filter uses voltage imputations in a different manner, depending on which stage of the filtering procedure is currently being applied. During the *prediction stage*, and if past voltage measurements are missing, the multiple-imputation algorithm suggests to draw voltage values from a proposal distribution ϕ. Each one of these imputations will define a different prior distribution for the next time instant, since $x_2(t+1)$ depends on $v(t)$ in Eq (19). However, as the transition equations place particles in different positions of the state-space, Rubin's rule of multiple imputation theory suggests that all those prior transition distributions should be combined into a single distribution by appropriate weighting (Rubin, 1987), yielding an increase in particle population.

Assuming that the prior distribution is known and the actual voltage value is unknown, then voltage imputations may also be considered for the *update stage* of the particle-filtering algorithm. Furthermore, in that case the resulting particles (which represent the posterior distribution) will keep the same location within the state-space. Thus, the number of particles is not increased since Rubin's rule (Rubin, 1987) is applied.

As multiple-imputed data generate an increase of the number of particles during the *prediction stage*, a *reduction stage* has to be incorporated into the algorithm to keep a fixed number of particles throughout time; avoiding a progressive increase of the particle population. This way, the SIR PF will work as it was originally designed, specially after voltage measurements are once more available.

The proposed MI PF implementation treats the problem of missing voltage observations, whereas the discharge current is assumed as an input variable known at each time instant. The imputation model adopted is defined as the probability distribution induced by Eq (20), providing prior knowledge about the voltage variability.

Denoting the multiple-imputed measurement data set as $\tilde{y}^j_{1:t} = \{\tilde{y}_{1:t-1}, y^j_t\}$ where $\tilde{y}_{1:t} = \{\tilde{y}_{1:t-1}, y^1_t, \ldots, y^m_t\}$, with $j \in \{1, \ldots, m\}$, the MI PF implementation is summarized in Algorithm 2.

4. EXPERIMENTAL RESULTS

In this article, the proposed multiple-imputation algorithm is applied to the case of SOC estimation in Li-Ion battery cells. Particularly, this method is intended to improve the way SOC is monitored on a BMS. A complete discharge cycle, containing a total of 2920 samples that were obtained from an experimental setup located at the Advanced Control Systems Laboratory, University of Chile, is analyzed for purposes of algorithm test and validation. To test the algorithm, a loss of 1000 sequential voltage measurements is considered. The estimation results are obtained with the use of 60 particles and 10 imputations. The performance is analyzed considering the average of 30 realizations for three different cases: i) SIR PF with a complete data set, ii) 1000-step prediction procedure along the missing measurements, and iii) MI PF with an incomplete data set.

Algorithm 2 Multiple Imputation Particle Filter

1. MI Importance Sampling
  if y_t is available **then**
    • SIR PF
  else
    for $j' = 1, \ldots, m$ **do**
      for $i = 1, \ldots, N$ **do**
        • Sample $\tilde{x}_t^{(i,j')} \sim \pi(x_t | x_{0:t-1}^{(i)}, \tilde{y}_{t-1}^{j'})$ and set $\tilde{x}_{0:t}^{(i,j')} \triangleq (x_{0:t}^{(i)}, \tilde{x}_t^{(i,j')})$
      end for
    end for
    • Compute m imputations $y_t^j \sim \phi(\{\tilde{x}_t^{(i,j')}, w_t^{(i,j')}\}, \eta_t)$ and its associated weights p_t^j.
    • Reduce the particle population from $N \cdot m$ to N.
    $\tilde{x}_t^{(i,j)} \rightarrow \tilde{x}_t^{(i)}$
    for $j = 1, \ldots, m$ **do**
      for $i = 1, \ldots, N$ **do**
        • Compute the importance weights
        $w_t^{(i,j)} = w_{t-1}^{(i)} \cdot \frac{p(y_t^j | \tilde{x}_t^{(i)}) \cdot p(\tilde{x}_t^{(i)} | x_{t-1}^{(i)})}{\pi(\tilde{x}_t^{(i)} | \tilde{x}_{0:t-1}^{(i)}, \tilde{y}_{1:t}^j)}$
        • Apply Rubin's rule
        $w_t^{(i)} = \sum_{j=1}^{m} w_t^{(i,j)}$
        • Normalize
        $\mathcal{W}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$
      end for
    end for
  end if

The probability density that was used in this case to draw voltage imputations corresponds to the distribution induced by Eq (20), where particles are obtained from the prior transition PDF shown in Eq. (18)-(19). The imputations are randomly drawn from the aforementioned distribution, and hence their weights are assumed to be equal.

In particular, the problem of reducing the number of particles from $N \cdot m$ to N -where N is the size of the original particle population and m is the number of imputations- could be achieved by resampling. However, this kind of technique fails because of the tendency to retain high probability particles only, discarding the uncertainty characterization provided by the MI strategy. Therefore, a suboptimal solution is proposed. The main focus consists on preserving the probability distribution described by $N \cdot m$ particles using only N of them. Thus, as an attempt to solve this problem, the particles are arranged as a function of the SOC ($\{x_k^{(l)}, \mathcal{W}_k^{(l)}\}_{l=1}^{N \cdot m}$) and clustered into groups of m particles, noting that the SOC corresponds to a state and its dynamic is described in Eq (19). One particle is obtained from each group by a weighted sum and its probability is assumed to be the sum of probabilities of each particle in the group. Therefore, the N new particles are generated as

$$\bar{\mathcal{W}}_k^{(i)} = \sum_{l=m\cdot(i-1)+1}^{m\cdot i} \mathcal{W}_k^{(l)} \quad (21)$$

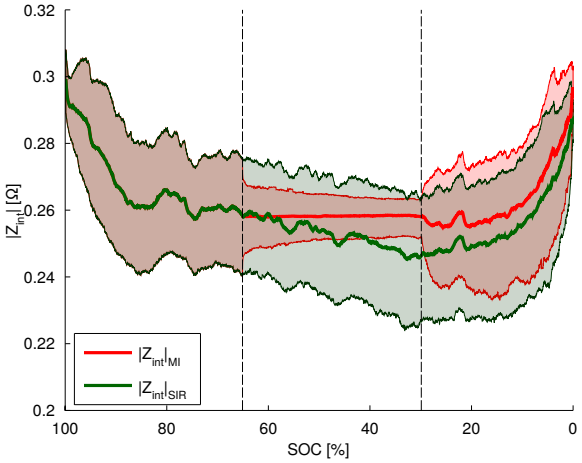
$$\bar{x}_k^{(i)} = \frac{1}{\bar{\mathcal{W}}_k^{(i)}} \cdot \sum_{l=m\cdot(i-1)+1}^{m\cdot i} \mathcal{W}_k^{(l)} x_k^{(l)} \quad (22)$$

$\forall\ i \in \{1, \ldots, N\}$. The biggest assumption adopted for the reduction stage was that the internal impedance remains constant at least when the battery's SOC is over 20%, which in practice makes it almost independent of the SOC. Of course, other factors also affect the value of the internal impedance, for example the battery temperature. In fact, that is the main reason why this parameter has to be estimated from voltage and discharge current measurements. The impact of these factors will not be considered in this particular version of the algorithm, but they will be included as part of future research work.

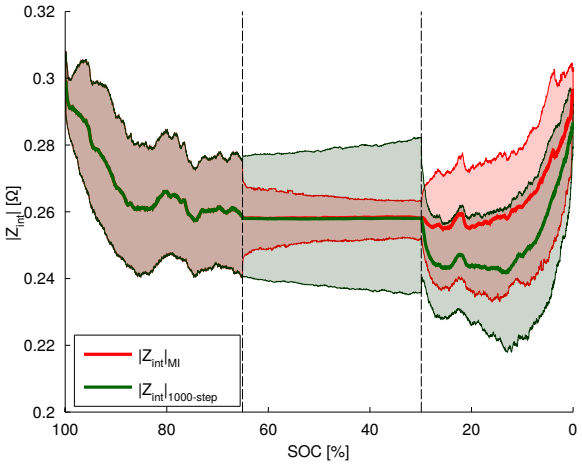
For this case study, the conventional SIR PF is applied in all the cases as long as there are no missing measurements. The focus lays on comparing the MI strategy to a simple n-step ahead prediction algorithm (Orchard et al., 2012) that could be applied when voltage measurements are lost. Also, the MI strategy will be compared to the PF-based estimates that are obtained with no missing data. Both comparisons yield results for internal impedance, SOC and voltage which are exposed in Figures 1, 2 and 3, respectively. For a better analysis, the same conditions are adopted for all the cases up to the time where data starts being lost.

As it is shown in Figure 1a, the assumption of a constant value for the internal impedance becomes invalid along the missing voltage window as the MI PF estimates differ significantly from a complete data estimation, out bounding the confidence intervals depicted by the MI PF estimation. In contrast, Figure 1b shows that the MI PF estimates are very similar to that of the 1000-step prediction but the uncertainties in this case differ among themselves mainly due to the hypothesis in the reduction stage of the MI PF, that leads to a uncertainty dimishment.

Regardless of what it has been mentioned before, the main feature of the proposed MI PF is ensuring a robust and bounded uncertainty characterization for the SOC, which is visualized in Figure 2. Figure 2a shows how the MI PF uncertainty overlaps that of the SIR PF whereas in Figure 2b this last is slightly overlapped by the uncertainty of the 1000-step prediction. It is interesting to note the MI strategy avoids the use of a resampling stage, yielding similar results as a long term prediction. Nevertheless, when voltage measurements are not lost anymore, a bias is added in both cases (MI PF and 1000-step prediction). This problem is generated by the approximately constant estimation for the internal impedance, which

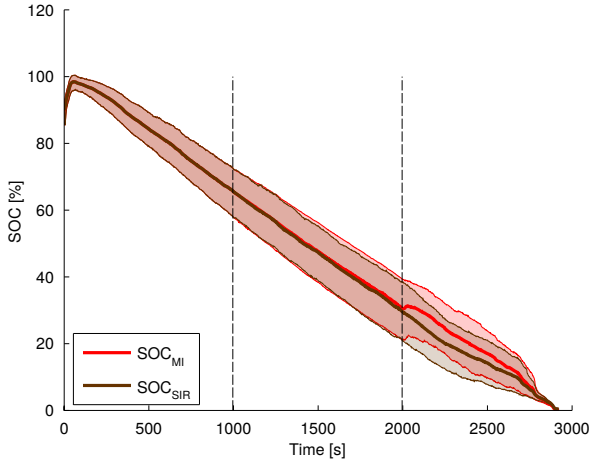


(a) MI and SIR PF

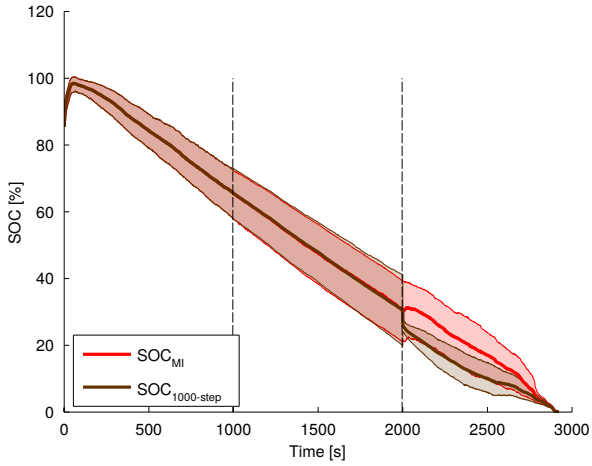


(b) MI PF and Prediction

Figure 1. Internal impedance estimation as a function of the SOC[%] for a disconnection of 1000 sequential voltage measurements denoted in the area between the dashed vertical lines. a) Comparison between the MI PF (red line) and the SIR PF (green line) with 95% confidence intervals. b) Comparison between the MI PF (red line) and the 1000-step prediction algorithm (green line) with 95% confidence intervals.



(a) MI and SIR PF



(b) MI PF and Prediction

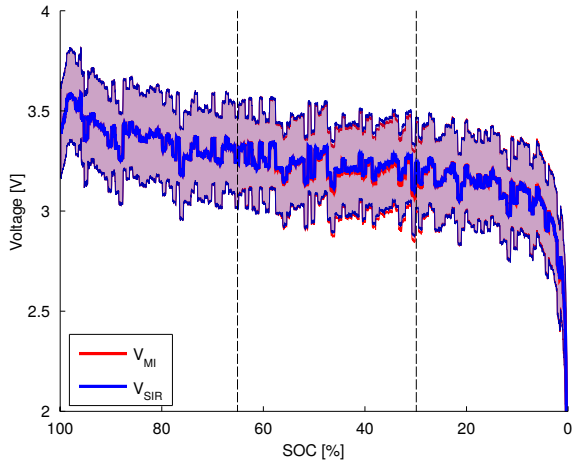
Figure 2. SOC estimation as a function of time[s] for a disconnection of 1000 sequential voltage measurements denoted in the area between the dashed vertical lines. a) Comparison between the MI PF (red line) and the SIR PF (brown line) with 95% confidence intervals. b) Comparison between the MI PF (red line) and the 1000-step prediction algorithm (brown line) with 95% confidence intervals.

introduces a bias affecting the SOC estimation as an attempt to correct the first. Notwithstanding, the uncertainty about the actual value of the internal impedance for the 1000-step prediction affects more intensively its performance when voltage is measured again than that of the MI PF. Consequently, the MI PF approach is the one who experiences better performance.

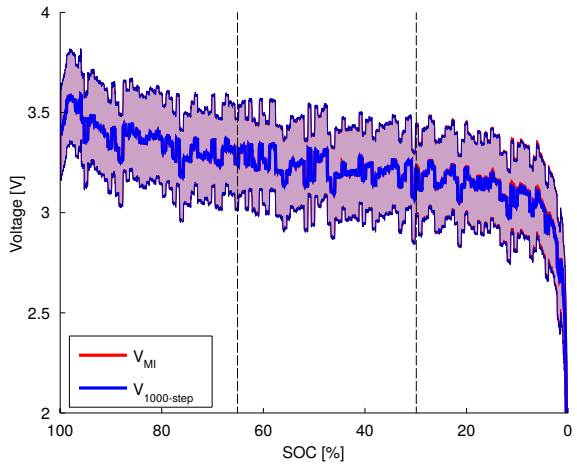
The underlying importance of holding a bounded uncertainty characterization on an estimation stage is that of providing appropiate conditions for a prognosis stage. Simultaneously, this converges into an improved performance of prognostic results due to a bounded uncertainty along the prediction horizon, hence predictions are obtained with a higher degree of certainty.

In the case of voltage estimation, the results are shown in Figure 3. Figure 3a shows that a bias is added to the voltage distribution corresponding to the MI PF. Note that it is obtained from using an imputation model based on the measurement model. The use of a few imputations (10 in this case study) provides a reasonable characterization of the output variability by generating a robust approximation to the true statistics even when data is partially lost. The bias remains negligible considering that the total amount of lost data reaches 1000. However, Figure 3b shows that the 1000-step

prediction shares its behavior, by describing nearly identical curves.



(a) MI and SIR PF



(b) MI PF and Prediction

Figure 3. Voltage estimation as a function of the SOC[%] for a disconnection of 1000 sequential voltage measurements denoted in the area between the dashed vertical lines. a) Comparison between the MI PF (red line) to the SIR PF (blue line) with 95% confidence intervals. b) Comparison between the MI PF (red line) and the 1000-step prediction algorithm (blue line) with 95% confidence intervals.

5. CONCLUSION

A new multiple-imputation particle-filtering based scheme for estimation when lost measurements are present is proposed where the Multiple Imputation Theory is the main core for uncertainty characterization. A particular implementation for SOC estimation is presented when voltage measures are sequentially lost along a period of time. Preliminary results show the success of the methodology by incorporating uncertainty by increasing the original number of particles, but then adding a reduction stage. However, a bias is added to the estimation process.

The case study for testing the algorithm includes a missing data window when the SOC is over a 20% of the battery's capacity. This allows the adoption of a simplified way for reducing particles in the algorithm based on the hypothesis that the value of the internal impedance remains constant. The MI strategy is compared to the case without missing data and also to a particle-filtering-based prognosis algorithm for performing a 1000-step prediction. The results show that the uncertainty characterization associated to the estimation stage -once the capacity to acquire data is no longer lost- is more appropiate if the MI PF is used than if the 1000-step prediction is used.

As the MI has been developed for offline applications, there are several aspects to consider for online applications. Some of them include improvements on the imputation model, adaptive estimation for an optimal number of particles and amount of imputations, alternative reduction methods of particle population, better ways for characterizing the internal impedance evolution in time, risk assessment, among others. Furthermore, the development of an optimal particle reduction may enable the connection of asynchronous networks, treatment for missing measurements, and prognosis, to give some examples.

ACKNOWLEDGMENTS

The authors would like to thank FONDECYT 1140774 and University of Costa Rica (Grant for Doctoral Studies) for their financial support.

REFERENCES

Andrieu, C., Doucet, A., & Punskaya, E. (2001). *Sequential monte carlo methods in practice* (A. Doucet, N. de Freitas, & N. Gordon, Eds.). Springer-Verlag.

Candy, J. (2009). *Bayesian signal processing: Classical, modern and particle filtering methods*. Wiley.

Crisan, D., & Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, *50*(3), 736-746.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, *10*(2), 197-208.

Graham, J., Olchowski, A., & Gilreath, T. (2007). How many imputations are really needed? some practical clarifications of multiple imputation theory. *Prevention Science*, *8*, 206-213.

Housfater, A., Zhang, X., & Zhou, Y. (2006). Nonlinear fusion of multiple sensors with missing data. *IEEE International Conference on Acoustics, Speech and Signal*

Processing, *4*, 961-964.

Liu, J., Kong, A., & Wong, W. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, *89*(425), 278-288.

Orchard, M., Cerda, M., Olivares, B., & Silva, J. (2012). Sequential monte carlo methods for discharge time prognosis in lithium-ion batteries. *International Journal of Prognostics and Health Management*, *3*, 1-12.

Orchard, M., & Vachtsevanos, G. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, *31*, 221-246.

Pattipati, B., Sankavaram, C., & Pattipati, K. (2011). System identification and estimation framework for pivotal automotive battery management system characteristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *41*(6), 869-884.

Pola, D. (2014). *An improved prognosis strategy with temperature-dependent state space models for the analysis of the state-of-health and state-of-charge in lithium-ion batteries*. M.Sc. Tesis. Department of Electrical Engineering, Universidad de Chile.

Rubin, D. (1987). *Multiple imputation for nonresponse in surveys*. Wiley.

BIOGRAPHIES

B.Sc. David E. Acuña was born in Santiago, Chile. He received his B.Sc. degree in Electrical Engineering (2014) from Universidad de Chile. He currently is a Research Assistant at the Lithium Innovation Center (Santiago, Chile). His research interests include signal processing, nonlinear filtering, and prognostics and health management for energy storage devices.

Dr. Marcos E. Orchard is Associate Professor with the Department of Electrical Engineering at Universidad de Chile and was part of the Intelligent Control Systems Laboratory at The Georgia Institute of Technology. His current research interest is the design, implementation and testing of real-time frameworks for fault diagnosis and failure prognosis, with applications to battery management systems, mining industry, and finance. His fields of expertise include statistical process monitoring, parametric/non-parametric modeling, and system identification. His research work at the Georgia Institute of Technology was the foundation of novel real-time fault diagnosis and failure prognosis approaches based on particle filtering algorithms. He received his Ph.D. and M.S. degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He received his B.S. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile. Dr. Orchard has published more than 50 papers in his areas of expertise.

Dr. Jorge F. Silva is Assistant Professor at the Department of Electrical Engineering, University of Chile, Santiago, Chile. He received the Master of Science (2005) and Ph.D. (2008) in Electrical Engineering from the University of Southern California (USC). He is IEEE member of the Signal Processing and Information Theory Societies and he has participated as a reviewer in various IEEE journals on Signal Processing. Jorge F. Silva is recipient of the Outstanding Thesis Award 2009 for Theoretical Research of the Viterbi School of Engineering, the Viterbi Doctoral Fellowship 2007-2008 and Simon Ramo Scholarship 2007-2008 at USC. His research interests include: non-parametric learning; sparse signal representations, statistical learning; universal source coding; sequential decision and estimation; distributive learning and sensor networks.

M. Sc. Aramis Pérez is a Research Assistant at the Lithium Innovation Center (Santiago, Chile) and professor at the School of Electrical Engineering at the University of Costa Rica. He received his B.Sc. degree (2002) and Licentiate degree (2005) in Electrical Engineering from the University of Costa Rica. He received his M. Sc. degree in Business Administration with a General Management Major (2008) and also he is a M. Sc. candidate in Industrial Engineering from the same university. Currently he is a doctorate student at the Department of Electrical Engineering at the University of Chile under Dr. Marcos E. Orchard supervision. His research interests include parametric/non-parametric modeling, system identification, data analysis, machine learning and manufacturing processes.

Qualitative Event-Based Fault Isolation under Uncertain Observations

Matthew Daigle[1], Indranil Roychoudhury[2], and Anibal Bregon[3]

[1] *NASA Ames Research Center, Moffett Field, California, 94035, USA*
matthew.j.daigle@nasa.gov

[2] *SGT Inc., NASA Ames Research Center, Moffett Field, California, 94035, USA*
indranil.roychoudhury@nasa.gov

[3] *Department of Computer Science, University of Valladolid, Valladolid, Spain*
anibal@infor.uva.es

ABSTRACT

For many systems, automatic fault diagnosis is critical to ensuring safe and efficient operation. Fault isolation is performed by analyzing measured signals from the system, and reasoning over the system behavior to determine which faults have occurred, based on models of predicted faulty behavior. For dynamic systems, reasoning may be performed using qualitative analysis of the differences between measured signals and their predicted values, in which observations take the form of qualitative symbols. Such an approach is quick to isolate faults, but depends critically on correct generation of the qualitative symbols from the signals. In this paper, we develop an approach to qualitative event-based fault isolation for dynamic systems that is robust to incorrect qualitative observations. Observations are treated as uncertain, where multiple interpretations of an observation, each with its own probability, are considered. By interpreting observed symbols in a probabilistic manner, the approach degrades gracefully as the number of incorrectly-generated symbols increases. The approach is demonstrated on an electrical power system testbed, and experiments using real data obtained from the hardware demonstrate the improved fault isolation performance in the presence of incorrect symbol generation.

1. INTRODUCTION

For many systems, automatic fault diagnosis is critical to ensuring safe and efficient operation. Within fault diagnosis, the task of fault isolation is concerned with an analysis of observed behavior in order to determine which fault has occurred. In many approaches, observations are transformed into a discrete symbolic (e.g., qualitative) form over which reasoning can be performed (Puig, Quevedo, Escobet, & Pulido, 2005; Koscielny & Zakroczymski, 2000). For dynamic systems, these discrete observations take the form of events (Daigle, Koutsoukos, & Biswas, 2009).

In qualitative fault isolation, residual signals are computed as the differences of observed behavior and predicted nominal behavior (Mosterman & Biswas, 1999). Deviations of the residual signals are then abstracted into symbolic, qualitative representations, called fault signatures, to facilitate diagnostic reasoning (specifically, +, −, and 0 symbols, representing increase, decrease, and no change from nominal, respectively). Fault models describe the potential sequences of fault signatures produced by faults, forming a qualitative event-based fault isolation approach (Daigle et al., 2009). Such an approach is quick to isolate faults, but depends critically on correct generation of these qualitative fault signatures. When the transformation from observed quantitative signals into observed qualitative fault signatures does not produce the correct result, the wrong information will be used to isolate faults, and this incorrect signature generation will, therefore, lead to incorrect diagnoses.

In this paper, we develop an *observation-robust* approach to qualitative event-based fault isolation for dynamic systems as an extension and generalization of the approach in (Daigle et al., 2009). Here, observation-robust means that the approach is still successful, to some degree, when encountering incorrect observations (henceforth, by *observation* we mean the version of the quantitative signal transformed into a qualitative symbol). By considering the qualitative observations as uncertain, and interpreting them in a probabilistic manner, the approach degrades gracefully as the number of incorrectly-generated symbols increases. The approach is

Matthew Daigle et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

demonstrated on the Advanced Diagnostics and Prognostics Testbed (ADAPT) (Poll et al., 2007) an electrical power system testbed that has served as a benchmark diagnostic system in the diagnostics community (Poll et al., 2011; Sweet, Feldman, Narasimhan, Daigle, & Poll, 2013). Using real experimental data obtained from the ADAPT hardware, we demonstrate the improved fault isolation performance in the presence of incorrect symbol generation.

Several previous works have used probabilistic solutions for different tasks of the fault diagnosis problem. In (Ricks & Mengshoel, 2009) the authors use Bayesian Networks (BNs) to represent probabilistic multi-variate models, which are applied to the ADAPT hardware, as we do in this paper. Other works have also applied BNs or Dynamic BNs (DBNs) for fault diagnosis, e.g., in (Pernestål, 2009) the author uses DBNs to improve the diagnosis of automotive vehicles, and in (Alonso-Gonzalez, Moya, & Biswas, 2011; Roychoudhury, 2009; Roychoudhury, Biswas, & Koutsoukos, 2010) DBNs are used for fault diagnosis. In all these cases, the probabilistic solutions are used to model the systems under conditions of uncertainty and then to perform diagnosis. However, more sources of uncertainty appear in the fault diagnosis process due to, for example, improper threshold selections or incorrect symbol generation. Our approach in this paper uses a model based on physical equations of the system, and performs fault diagnosis using this model. The probabilistic methods are then used to reduce the uncertainty in fault isolation due to incorrectly-generated symbols. An approach similar to our work is presented in (Ying, Kirubarajan, Pattipati, & Patterson-Hine, 2000), in the sense that a probabilistic solution is used to perform fault diagnosis in systems with imperfect diagnosis tests. However, the diagnosis approach and the probabilistic solution are different than those used in this paper.

The remainder of the paper is organized as follows. Section 2 formulates the problem for event-based fault isolation. Section 3 reviews the standard event-based fault isolation approach, and Section 4 extends the approach to be observation-robust. Section 5 describes implementations of the standard and robust frameworks based on qualitative fault isolation, and presents the case study and results. Section 6 concludes the paper and discusses future work.

2. PROBLEM FORMULATION

In this section, we define the fault isolation problem that we aim to solve. We assume an event-based fault isolation framework, where faults are isolated based on the analysis of a sequence of observable events produced as a result of the fault occurrence (where, in the nominal case, no such events are produced). The approach is related to discrete-event diagnosis (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1996) and, more closely, the concept of chronicles (Cordier & Dousson, 2000). For the purposes of defining the problem and describing the fault isolation approach, we present a generalized theoretical framework for event-based fault isolation. In Section 5, we will describe a specific implementation of this framework for dynamic systems (Daigle et al., 2009).

First, we have the set of faults, F, that may occur in the system. Faults produce observable events, called *fault signatures*.

Definition 1 (Fault Signature)**.** A *fault signature* for a fault f denoted by σ_f, is an event that is observed as a consequence of the occurrence of f. The set of fault signatures for f is denoted as Σ_f. The set of fault signatures over a set of faults F is denoted as Σ_F, i.e., $\Sigma_F = \bigcup_{f \in F} \Sigma_f$.

These events are produced in some temporal order. A *fault trace* is a one particular fault signature sequence that may be observed.

Definition 2 (Fault Trace)**.** A *fault trace* for a fault f denoted by λ_f, is a sequence of fault signatures from Σ_f resulting from the occurrence of f.

Definition 3 (Maximal Fault Trace)**.** A fault trace λ_f for a fault f is *maximal* if there is no extension $\lambda_f \sigma_f$ that is also a fault trace for f.

The set of all possible maximal fault traces for a fault is called its *fault language*.

Definition 4 (Fault Language)**.** The *fault language* of a fault $f \in F$ denoted by L_f, is the set of all maximal fault traces for f. The union of fault languages for a set of faults F is denoted as L_F, i.e., $L_F = \bigcup_{f \in F} L_f$.

We assume that we have considered all possible faults in F, and that the fault languages are complete.

Assumption 1 (Completeness of F)**.** We assume that F is complete, i.e., there is no other fault $f \notin F$ that can occur.

Assumption 2 (Completeness of L_f)**.** We assume that for every fault $f \in F$, L_f is complete, i.e., there is no other maximal fault trace $\lambda_f \notin L_f$ that may occur as a result of f.

By Assumptions 1 and 2, whenever some fault trace λ occurs, it must have been produced by some fault $f \in F$, and it must belong to L_f for at least one $f \in F$. These assumptions are quite standard in model-based diagnosis. In some approaches, e.g., (Hofbaur & Williams, 2002; Narasimhan & Brownston, 2007), an *unknown fault* is considered, which is consistent with everything. In our approach, such a fault could be included by adding a new f where L_f contains all possible traces.

So, associated with each fault is a set of fault traces, where the maximal fault traces are collected into a fault language. When a fault occurs, a specific event sequence will be observed that belongs to the fault language. In this framework,

Algorithm 1 $F^* \leftarrow$ `FaultIsolation`(F)

1: $F^* \leftarrow F$
2: $\lambda \leftarrow \varnothing$
3: **while** σ_i **observed do**
4: $\quad \lambda \leftarrow \lambda\sigma_i$
5: $\quad F^* \leftarrow$ `FindConsistentFaults`(F^*, λ)
6: **end while**

fault isolation reduces to matching observed fault traces to predicted fault traces, to determine which fault has occurred. So, the fault isolation problem is defined as follows.

Problem. Given an observed fault trace, λ, find the *most likely* single fault f that produced λ.

Here, we aim to find the *most likely* fault, because the observed fault trace may not always be generated correctly, due to various reasons, such as improperly tuned quantitative signal thresholds. If this is the case, we must find the most likely fault that explains the (incorrectly) observed trace, because the observed trace may not be found in any L_f. The standard fault isolation approach (Section 3) assumes the observed trace is always correct, whereas the new robust approach (Section 4) does not make that assumption, in order to handle incorrectly observed fault traces in a robust fashion.

3. EVENT-BASED FAULT ISOLATION

In the standard fault isolation approach, we assume that fault traces are correctly observed.

Assumption 3. All observed fault signatures are correct, i.e., if fault signature σ occurs, it is observed as σ.

Therefore, given Assumptions 1–3, when a fault occurs and we observe a fault trace, this trace must belong to the fault language of at least one fault. The function of the fault isolation algorithm is simply to find which faults are consistent with the observed fault trace.

The fault isolation algorithm is presented as Algorithm 1. Initially, the set of isolated faults, F^*, is set to the complete set of faults, F. The initial observed fault trace λ is the empty event sequence. While new fault signatures are observed, we update the observed fault trace, and reduce F^* to the set of faults consistent with the new trace.

The `FindConsistentFaults` algorithm, presented as Algorithm 2, eliminates from F^* faults that are no longer consistent with the trace extended with σ_i. A fault f is consistent with an observed trace λ if there is a fault trace λ_f in its fault language where λ is a prefix ($\sqsubseteq$), i.e., the fault can generate the observed sequence of events so far. If the fault is indeed consistent, it is retained, otherwise, it is removed from F^*.

Basically, we continue to observe new symbols, and F^* reduces. If the system is *diagnosable*, i.e., all faults are distinguishable from each other (via their fault languages), then F^* will reduce to a single fault. A fault f_i is distinguishable from fj in this framework if there is no trace in $\mathcal{L}_{f_i}$ that is a prefix of a trace in $\mathcal{L}_{f_j}$.

Algorithm 2 $F^* \leftarrow$ `FindConsistentFaults`(F^*, λ)

1: **for all** $f \in F^*$ **do**
2: $\quad$ **if** $\neg$ **exist** $\lambda_f \in L_f$ **such that** $\lambda \sqsubseteq \lambda_f$ **then**
3: $\quad\quad F^* \leftarrow F^* - \{f\}$
4: $\quad$ **end if**
5: **end for**

Example 1. Consider a set of three faults, $F = \{f_1, f_2, f_3\}$, where $L_{f_1} = \{cab, acb\}$, $L_{f_2} = \{abc, bac\}$, and $L_{f_3} = \{cb, ca, ab\}$. Say that we observe first the fault signature a. Each of the faults may produce a as the first fault signature, so $F^* = \{f_1, f_2, f_3\}$. Say we next observe b. Now, f_1 cannot produce a trace starting with ab, so it is eliminated, and $F^* = \{f_2, f_3\}$. Say we next observe c. Now, f_3 cannot produce a trace beginning with abc, and so f_2 is isolated as the fault.

Let us say we observe a trace that does not belong to any fault language. There are three explanations for this: (*i*) an unknown fault has occurred (violation of Assumption 1), (*ii*) a valid trace is missing from a fault language (violation of Assumption 2), or (*iii*) the trace was observed incorrectly (violation of Assumption 3). For (*i*) and (*ii*), there is nothing that can be done, so we limit ourselves only to situation (*iii*). So, what happens when the trace is observed incorrectly?

Example 2. Consider again the fault set from the previous example. Say we observe c, then we have $F^* = \{f_1, f_3\}$. Say we then observe b, then we have $F^* = \{f_3\}$. Say we then observe a, then we have $F^* = \varnothing$, i.e, all faults were eliminated. One explanation is that the a fault signature was falsely observed (i.e., a false alarm), in which case the true fault is f_3.

The result of an incorrectly observed trace is an incorrect fault isolation result. Either all candidates will be eliminated, as in the example above, or the wrong fault will be isolated (if the observed trace belongs to a fault language of a fault that did not occur). In practice, it is not unlikely that a trace may be incorrectly observed, e.g., from noisy sensor signals, overly sensitive fault detection thresholds, etc. Clearly, Algorithm 1 is not robust in this case. A more robust approach is necessary to handle a violation of Assumption 3.

4. ROBUST EVENT-BASED FAULT ISOLATION

As described in Section 3, Algorithm 1 makes Assumption 3, i.e., there is only one interpretation of an observed trace, which is what was observed. In practice, however, traces may be incorrectly observed, and so we must drop Assumption 3 in order to be robust to this situation, i.e., to make the approach *observation-robust*. In more detail, by observation-robust, we mean that the approach performs optimally when all observations are correct, and its performance degrades gracefully as the number of incorrect observations increases.

In practical terms, this means that the true fault is diagnosed to have the highest probability of being the one that occurred, when all observations are correct. Further, its assigned probability decreases when incorrect observations are encountered, where, up to a certain point, it remains the most probable fault given the observations.

In order to still perform in the face of incorrect observations, we must differentiate between an *observed trace* and an *interpreted trace*. For a given observed trace, there are several potential interpreted traces. An observed trace may or may not belong to any L_f. Any valid interpretation of it, however, must be a prefix of some trace in L_F. That is, given an observed trace, we must generate all correct ways to interpret it, given the set of considered faults. Each interpreted trace will have its own probability and its own diagnosis. Given the set of interpreted traces, their probabilities, and their diagnoses, we can extract a combined diagnosis that provides, for every fault resulting from an interpreted trace, a probability of its occurrence.

Say that so far we have an interpreted trace of λ, and a new symbol σ_i is observed. How do we extend λ given σ_i? We assume there is a known set of signatures, Σ_{σ_i}, that can be observed as σ_i. At a minimum, this set contains σ_i itself. So, when σ_i is observed, it could have been any signature in Σ_{σ_i} that actually occurred. However, only a subset of these can extend λ and be consistent with a given set of faults. To be consistent, they have to be a prefix of some trace found in L_F (since an interpreted trace must belong to L_F).

Example 3. Consider again the set of three faults, $F = \{f_1, f_2, f_3\}$, where $L_{f_1} = \{cab, acb\}$, $L_{f_2} = \{abc, bac\}$, and $L_{f_3} = \{cb, ca, ab\}$. Say that $\Sigma_a = \{a, b\}$, $\Sigma_b = \{b, a\}$, and $\Sigma_c = \{c\}$. Say that the trace bca is observed, what are the possible interpreted traces? First b is observed and that can be interpreted as either a or b; so far the interpreted traces are a and b. Next c is observed, which can be interpreted only as c; so the interpreted traces are ac and bc. Then a is observed, which can be interpreted as either a or b, so the potential interpreted traces are aca, acb, bca, bcb, however, only acb belongs to a fault language and is valid.

Σ_{σ_i} may also contain special signatures that represent false alarms, which we denote using ϵ with a subscript denoting the event associated with the false alarm (e.g., ϵ_a for a false alarm of event a). For example, we could observe some signature σ, but it may be possible that no signature occurred and σ is to be interpreted as a false alarm. In this case, we require a special false alarm signature. The fault languages must include traces that contain false alarm signatures in order for them to be interpreted from an observed trace. Note that such signatures are not required for the standard approach due to Assumption 3. We require also a false alarm "fault" to be included in F, for which its traces contain only false alarm signatures. It is not actually a fault but used to represent the situation where so far, only false alarm signatures have been interpreted from the observed signatures.

Example 4. Consider the same situation as in the previous example, except with false alarm signatures ϵ_a, ϵ_b, and ϵ_c. The fault languages are extended by traces where a, b, and c can be replaced with these signatures, respectively, e.g., L_{f_1}, in addition to cab, has $\epsilon_c ab$, $c\epsilon_a b$, and $ca\epsilon_b$, as well as $\epsilon_a cb$, $\epsilon_b ca$, $\epsilon_a \epsilon_b c$, etc. Here, we have $\Sigma_a = \{a, b, \epsilon_a\}$, $\Sigma_b = \{b, a, \epsilon_b\}$, and $\Sigma_c = \{c, \epsilon_c\}$. We require then also the false alarm fault E, which has all traces of the three signatures ϵ_a, ϵ_b, and ϵ_c. Say again that the trace bca is observed, what are the possible interpreted traces? First b is observed and that can be interpreted as either a, b, or a false alarm in b, ϵ_b. Then c is observed which is really either c or ϵ_c, so the potential interpreted traces are ac, $a\epsilon_c$, $b\epsilon_c$, $\epsilon_b c$, $\epsilon_b \epsilon_c$ (bc is not included since it does not belong to any fault language). Next a is observed which is either a, b, or ϵ_a. The interpreted traces are then acb, $a\epsilon_c b$, $b\epsilon_c a$, $b\epsilon_c \epsilon_a$, $\epsilon_b ca$, $\epsilon_b c\epsilon_a$, $\epsilon_b \epsilon_c a$, and $\epsilon_b \epsilon_c \epsilon_a$.

The algorithm for robust fault isolation is given as Algorithm 3. We keep a set of tuples, $\mathcal{L}$, containing an interpreted trace λ, its probability p, and its diagnosis F^*. Initially, the set contains only one tuple, which is the empty trace ε, with a probability of 1 and the complete fault set F as its diagnosis. When a new signature σ_i is observed (ln. 2), we go through each interpreted trace λ. First, we find all new signatures that would (*i*) belong to Σ_{σ_i}, and (*ii*) can extend λ to produce a valid fault trace (ln. 5). For each of these possible next signatures, we extend the trace with it (ln. 7), assign the new trace's probability (lns. 8–15), and obtain its diagnosis (ln. 16). We then add the new tuple (λ', p', F^*) to the set of new tuples $\mathcal{L}'$ (ln. 17), which replaces $\mathcal{L}$ (ln. 20). Finally, we construct the merged diagnosis $\mathcal{F}^*$, which is a set of tuples of a fault and its probability.

To compute the probability of a trace, we assume that there is a probability of observing the correct signature, p_c. We can compute the probability of the interpreted signature, p_σ, as p_c if it matches the observed signature σ_i. If it does not match, we assume that all other signatures are equally probable, so it is assigned as $(1 - p_c)/(|\Sigma| - 1)$ if σ_i is possible to observe, and $1/|\Sigma|$ if not. The probability of the trace extended by σ is then the probability of the original trace times the probability of σ.

The diagnosis that is merged over all traces is computed as described in Algorithm 4. Each fault is assigned initially a probability of 0. Then, for each interpreted trace, the probability of the fault given that trace, $p(f|\lambda)$, is computed as the a priori probability of the fault divided by the sum of the probabilities of that fault diagnosed for that trace. This probability is then added to the probability of the fault, $p(f)$. After going through all traces, each fault is assigned its total probability. The set $\mathcal{F}^*$ is created by adding tuples for all faults and their probabilities.

Algorithm 3 $\mathcal{F}^* \leftarrow$ `RobustFaultIsolation`(F)

1: $\mathcal{L} \leftarrow \{(\varepsilon, 1, F)\}$
2: **while** σ_i **observed do**
3: $\quad\mathcal{L}' \leftarrow \varnothing$
4: $\quad$**for all** $(\lambda, p, F^*) \in \mathcal{L}$ **do**
5: $\quad\quad\Sigma \leftarrow \{\sigma : \sigma \in \Sigma_{\sigma_i}$ **and exists** $\lambda \in L_{F^*}$ **such that** $\lambda\sigma \sqsubseteq \lambda\}$
6: $\quad\quad$**for all** $\sigma \in \Sigma$ **do**
7: $\quad\quad\quad\lambda' \leftarrow \lambda\sigma$
8: $\quad\quad\quad$**if** $\sigma = \sigma_i$ **then**
9: $\quad\quad\quad\quad p_\sigma \leftarrow p_c$
10: $\quad\quad\quad$**else if** $\sigma_i \in \Sigma$ **then**
11: $\quad\quad\quad\quad p_\sigma \leftarrow (1 - p_c)/(|\Sigma| - 1)$
12: $\quad\quad\quad$**else**
13: $\quad\quad\quad\quad p_\sigma \leftarrow 1/|\Sigma|$
14: $\quad\quad\quad$**end if**
15: $\quad\quad\quad p' \leftarrow p \cdot p_\sigma$
16: $\quad\quad\quad F^* \leftarrow$ `FindConsistentFaults`(F^*, λ')
17: $\quad\quad\quad\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(\lambda', p'F^*)\}$
18: $\quad\quad$**end for**
19: $\quad$**end for**
20: $\quad\mathcal{L} \leftarrow \mathcal{L}'$
21: $\quad\mathcal{L} \leftarrow$ `Prune`$(\mathcal{L})$
22: $\quad\mathcal{F}^* \leftarrow$ `ConstructF`$(F, \mathcal{L})$
23: **end while**

Algorithm 4 $\mathcal{F}^* \leftarrow$ `ConstructF`$(F, \mathcal{L})$

1: $\mathcal{F}^* \leftarrow \varnothing$
2: **for all** $f \in F$ **do**
3: $\quad p(f) \leftarrow 0$
4: **end for**
5: **for all** $(\lambda, p, F^*) \in \mathcal{L}$ **do**
6: $\quad$**for all** $f \in F^*$ **do**
7: $\quad\quad p(f|\lambda) \leftarrow \dfrac{p_f}{\sum_{f' \in F^*} p_{f'}}$
8: $\quad\quad p(f) \leftarrow p(f) + p \cdot p(f|\lambda)$
9: $\quad$**end for**
10: **end for**
11: **for all** $f \in F$ **do**
12: $\quad\mathcal{F}^* \leftarrow \mathcal{F}^* \cup \{(f, p(f))\}$
13: **end for**

Clearly, the number of interpreted traces, in the worst case, grows exponentially with each new observed symbol. Each new symbol can be interpreted in a number of ways and all current interpreted traces need to be extended with all possible interpretations. In order to control the computational complexity of the algorithm, a pruning step is added (ln. 21). Interpreted traces may be removed from $\mathcal{L}$ by, for example, keeping only the N most probable traces, or keeping only traces above a probability threshold p_o. After removing traces from $\mathcal{L}$, the trace probabilities must be normalized.

Example 5. Consider again the scenario in the previous example. The diagnostic tree is shown in Fig. 1. Initially, any of the faults are possible, including the false alarm fault E. The branches in the tree represent the possible interpreted traces from the observed trace bca. The standard approach would have only one branch. We assume that $p_c = 0.9$, and the arrows are labeled with the interpreted symbol and its probability, leading to the new diagnosis and its probability. Since bca does not belong to any fault language, the standard approach would fail, whereas in this approach, we have many potential diagnoses that are ranked probabilistically, depending on the probabilities assigned to the interpreted symbols. For example, take the leftmost branch, where b is correctly observed. This happens with 90% probability, and immediately leads to $\{f_2\}$ as the diagnosis, since no other fault can produce a b as the first signature. Then c is observed. Since there is no fault that can produce bc, the only valid interpretation, given that b was correctly observed, is that c was incorrectly observed and the interpreted signature is ϵ_c, i.e., a false alarm of symbol c. Then a is observed, which can be interpreted only as a or ϵ_a, but not as b since no fault produces two b signatures in any trace. In either case, the diagnosis remains f_2. The rightmost branch, on the other hand, represents the case where all observations were false alarms, and thus the diagnosis is E. For a given fault, its total probability over all interpreted traces can be computed. If we assume that all faults are equally likely, then $p(f_2|bca) = 0.81 + 0.09 + 0.005/3 + 0.0045/3 = 0.9032$.

Clearly, the selection of values for p_c and p_o will determine the final computed probabilities of candidates for a given observed trace. A higher value of p_c will assign a higher probability to the most consistent candidates and a lower value to the remaining candidates, i.e., the candidate probability distribution will have a smaller variance. Similarly, a lower value of p_c will cause the candidate probability distribution to have a larger variance. If p_o is too high, and a trace is incorrectly observed, then it is possible that the correct candidate can be eliminated. Therefore, both p_c and p_o have to be selected to best represent the confidence in the symbol observation process.

5. CASE STUDY

In this section, we describe the application of the new robust event-based fault isolation framework to ADAPT. We use the qualitative event-based fault isolation (QFI) framework developed in (Daigle et al., 2009) and apply the robust methodology to it. We first describe the QFI framework and how it maps into the general event-based framework described earlier, then describe the ADAPT system. Finally, we describe experimental results using data from ADAPT.

5.1. Qualitative Event-Based Fault Isolation

In the QFI framework in (Mosterman & Biswas, 1999; Daigle et al., 2009), signatures capture qualitative deviations in magnitude and slope of residual signals, where a residual is computed as the difference between a measured value of a sensor and its expected (model-predicted) value. So, for a given residual r, we can have six different signatures: (*i*) an increase in magnitude, (*ii*) a decrease in magnitude, (*iii*) an increase in slope, (iv) a decrease in slope, (*v*) a false alarm in the magnitude, and (*vi*) a false alarm in the slope. For each potential fault, we can use a dynamic system model to determine

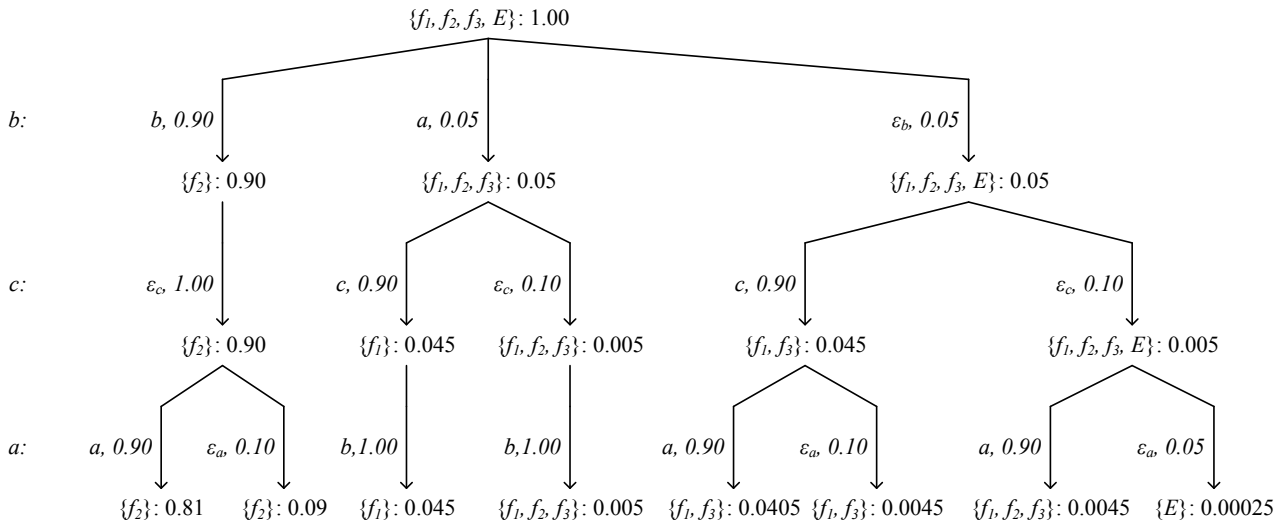


Figure 1. Example diagnostic tree.

which signatures are possible, as described in (Mosterman & Biswas, 1999).

Fault traces in this framework obey a certain set of constraints. First, for a given residual r, the magnitude symbol must always be observed before the slope symbol, and magnitude and slope symbols can be observed only once per residual (including false alarm signatures). Second, the order of signatures between residuals must respect relative residual orderings (Daigle, Koutsoukos, & Biswas, 2007), which express the intuition that faults manifest in some residuals before others. Like signatures, these can be derived from a dynamic system model (Daigle, 2008). Third, once a false alarm signature occurs for the magnitude, we cannot observe any more signatures for that residual. Aside from these restrictions, false alarms can occur at any time. In this framework, fault traces do not need to be precomputed but can be computed online (Daigle et al., 2009).

More information on this framework and its implementation may be found in (Daigle, Roychoudhury, & Bregon, 2013; Daigle, Bregon, & Roychoudhury, 2011). For the purposes of this paper, it suffices to say that we build a dynamic model in order to compute residuals, and these are analyzed in a statistical manner to generate observed signatures. This involves the use of thresholds on the residuals. The major practical problem here is tuning of the thresholds, which can be time-consuming in order to achieve the desired false alarm/missed detection trade-off. If these are not perfectly tuned, signatures can be incorrectly generated. In practice, this is quite difficult, so, using an approach that is robust to incorrect signatures is much desired. We compare two different diagnosers, (*i*) the QED algorithm, which implements the `FaultIsolation` algorithm; and (*ii*) probabilistic QED (pQED), which implements the `RobustFaultIsolation` algorithm. Except for the fault isolation algorithm, the two diagnosers are the same.

5.2. ADAPT

In this paper, we apply our new methodology to the Advanced Diagnostics and Prognostics Testbed (ADAPT), an electrical power distribution system that is representative of those on spacecrafts. ADAPT serves as a testbed through which faults can be injected to evaluate diagnostic algorithms (Poll et al., 2007). ADAPT has been established as a diagnostic benchmark system through the industrial track of the International Diagnostic Competition (DXC) (Kurtoglu et al., 2009; Poll et al., 2011; Sweet et al., 2013). In particular, this paper is focused on diagnosing faults on a subset of ADAPT, called ADAPT-Lite.

A system schematic for ADAPT-Lite is given in Fig. 2. A battery (BAT2) supplies electrical power to several loads, transmitted through several circuit breakers (CB236, CB262, CB266, and CB280) and relays (EY244, EY260, EY281, EY272, and EY275), and an inverter (INV2) that converts dc to ac power. ADAPT-Lite has one dc load (DC485) and two ac loads (AC483 and FAN416). There are sensors throughout the system to report electrical voltage (names beginning with "E"), electrical current ("IT"), and the positions of relays and circuit breakers ("ESH", "ISH"). Finally there is one sensor to report the operating state of a load (fan speed, "ST") and another to report the battery temperature ("TE"). Models and additional details for ADAPT-Lite can be found in (Daigle et al., 2011, 2013).

Our list of potential faults includes failures in the relays, circuit breakers, fan, DC load, and AC load. We consider also

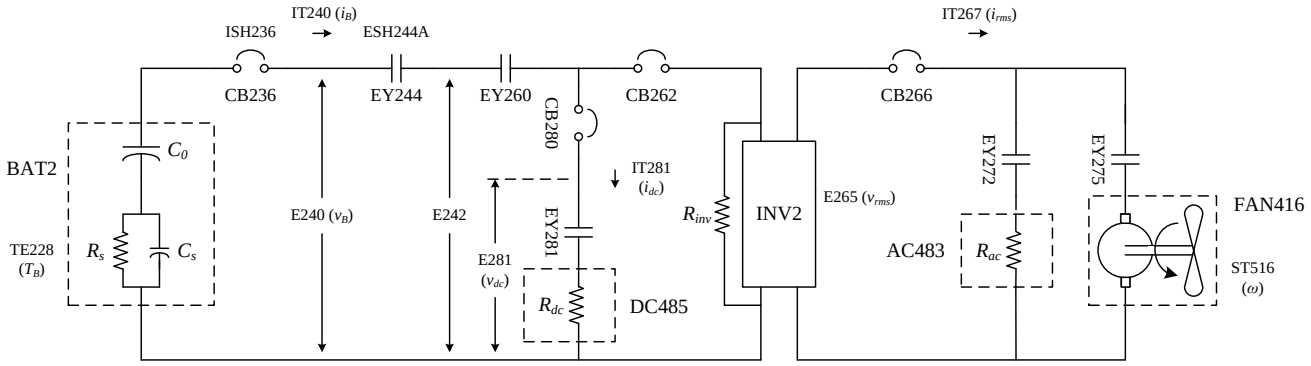


Figure 2. ADAPT-Lite schematic.

under- and over-speed faults of the fan, and offset, drift, and intermittent offset faults in the DC and AC loads.

5.3. Experiments

Using scenarios available from the DXC, we ran QED and pQED on a set of 30 nominal scenarios and 71 fault scenarios. The same fault detectors were used for both algorithms, so that we can show that, when incorrect signatures are generated, pQED performs better than QED, with the same information. The settings are nonoptimal in order to better highlight the differences in the approaches when multiple incorrect observations are encountered; improving the settings would of course improve the performance of both algorithms, but make it harder to compare the performance in nonoptimal conditions.

We first consider an example scenario, to illustrate the different diagnosis approaches. We then summarize the performance of the approaches over all scenarios.

As an example, consider a resistance drift fault in AC483. The fault is injected at 60 s and detected at 63 s with a decrease in IT240. QED reduces the candidate list to a failure in AC483, a positive resistance offset in AC483, a positive resistance drift in AC483, a failure in CB236, CB262, CB266, EY244, and DC485, a resistance increase in DC485, a resistance drift in DC485, a failure in EY244, EY260, EY272, EY275, EY284, FAN416, an under-speed fault in FAN416, and a failure in INV2. A − signature for the slope of the IT240 residual is then computed, for which only the drift faults are consistent. An increase in E242 is detected at 120 s, followed by the generation of a + signature for its slope. QED eliminates all faults, because it expects IT267 to deviate before E242. On the other hand, pQED retains the drift faults as candidates, but lowers their probabilities. Before the E242 deviation, the two drift faults had a probability of 38.77% each. After, the probability reduces to 3.92%, and they are still at the top of the candidate list. With the subsequent signatures for E242, probability decreases, as this is more evidence of other potential faults, but they remain the most probable. However, then E240 deviates, again before IT267 as expected, and this reduces their probability further, and they drop to the eighth and ninth most probable (at this point it is more likely that the detection of a negative slope (rather than no change in slope) was incorrect, and so failures in the circuit breakers and relays become more likely). In this case, no deviation was detected in IT267. With a more sensitive threshold, a deviation in IT267 could have been detected first, and the drift faults would have remained the most probable. Although this is not the most optimal result, at least the true fault was contained in the final diagnosis, albeit not at the highest level of probability.

5.3.1. Summary of Results

Over the nominal scenarios, both algorithms (since they use the same fault detectors) correctly detected a fault (true positives) 69 of 71 times, with 2 missed detections (false negatives). There were no false alarms detected.

For the fault scenarios, QED ends with a list of candidates that are consistent with the observed symbols. Ideally, this list is a singleton, containing the true fault. If, given the available diagnostic information, this is not possible, then we desire that it has the true fault in its final candidate list. In fact, QED never obtains the true fault as the single candidate, as diagnosability is not high enough to achieve that condition.

QED has the correct fault in its candidate list in 24 of 69 scenarios. This means that there are incorrect signatures generated in at least 45 scenarios. This can be improved with better fault detector tuning, however we keep these settings in order to demonstrate the improvement pQED provides. In 32 of these 45 scenarios, QED actually eliminates all faults, as no faults were consistent with the (incorrect) observations.

For pQED, we used $p_c = 90\%$, and pruned candidates with probability less than 0.1%. If pQED does not prune, then it will always have the correct candidate in its candidate list (but perhaps with a low probability assignment). With the pruning threshold used, pQED has the correct candidate in

its final list 63 of 69 times, which is a significant improvement over QED. For the 6 times in which it did not have the true fault, there were too many incorrect observations, bringing down the probability of the true fault low enough that all traces containing the fault were pruned.

Of course, it is not enough the pQED has the correct fault in its list, as this depends solely on the pruning threshold. We are interested in the probability assignment of the true fault within the final candidate list. pQED diagnoses the true fault as the fault with highest probability 38 of 69 times. This is better than the 24 of 69 times for QED. Since QED does not rank its final candidates, pQED's result is actually significantly better and more useful. For the times when the true fault is not ranked the highest, it is at least contained in the final candidate list for most of the time.

6. CONCLUSIONS

In this paper, we presented a robust approach to event-based fault isolation that drops the observation correctness assumption in order to improve robustness of fault isolation when events are incorrectly observed. We applied this framework to a qualitative event-based fault isolation framework. Experiments using real data from an electrical power system testbed demonstrated the approach and its improved robustness.

Future work will focus on extending the approach to multiple fault isolation, and extending the probability framework to account for conditional probabilities.

ACKNOWLEDGEMENTS

M. Daigle's and I. Roychoudhury's funding for this work was provided by the NASA System-wide Safety and Assurance Technologies (SSAT) Project. A. Bregon's funding for this work was provided by the Spanish MICINN DPI2013-45414-R grant.

REFERENCES

Alonso-Gonzalez, C., Moya, N., & Biswas, G. (2011). Dynamic bayesian network factors from possible conflicts for continuous system diagnosis. In *Proc. of the 14th int. conf. on advances in ai* (pp. 223–232). Berlin: Springer-Verlag.

Cordier, M.-O., & Dousson, C. (2000, June). Alarm driven monitoring based on chronicles. In *Proceedings of the 4th symposium on fault detection supervision and safety for technical processes* (p. 286-17291).

Daigle, M. (2008). *A qualitative event-based approach to fault diagnosis of hybrid systems*. Unpublished doctoral dissertation, Vanderbilt University.

Daigle, M., Bregon, A., & Roychoudhury, I. (2011, October). Qualitative Event-based Diagnosis with Possible Conflicts: Case Study on the Third International Diagnostic Competition. In *Proceedings of the 22nd international workshop on principles of diagnosis* (p. 285-292). Murnau, Germany.

Daigle, M., Koutsoukos, X., & Biswas, G. (2007, April). Distributed diagnosis in formations of mobile robots. *IEEE Transactions on Robotics*, *23*(2), 353–369.

Daigle, M., Koutsoukos, X., & Biswas, G. (2009, July). A qualitative event-based approach to continuous systems diagnosis. *IEEE Transactions on Control Systems Technology*, *17*(4), 780–793.

Daigle, M., Roychoudhury, I., & Bregon, A. (2013, October). Qualitative event-based diagnosis with possible conflicts: Case study on the fourth international diagnostic competition. In *Proceeedings of the 24th international workshop on principles of diagnosis* (p. 230-235).

Hofbaur, M., & Williams, B. (2002, May). Hybrid diagnosis with unknown behavioral modes. In *Proceedings of the 13th international workshop on principles of diagnosis* (pp. 97–105).

Koscielny, J., & Zakroczymski, K. (2000). Fault isolation method based on time sequences of symptom appearance. In *Proceedings of ifac safaprocess.* Budapest, Hungary.

Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., Kuhn, L., de Kleer, J., ... Feldman, A. (2009, June). First international diagnosis competition – DXC'09. In *Proceedings of 20th international workshop on principles of diagnosis* (p. 383-396).

Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *29*(6), 554-565.

Narasimhan, S., & Brownston, L. (2007, May). HyDE — a general framework for stochastic and hybrid model-based diagnosis. In *Proc. of the 18th int. workshop on principles of diagnosis* (pp. 162–169).

Pernestål, A. (2009). *Probabalistic fault diagnosis with automotive applications*. Unpublished doctoral dissertation, Linköping University.

Poll, S., de Kleer, J., Abreau, R., Daigle, M., Feldman, A., Garcia, D., ... Sweet, A. (2011, October). Third international diagnostics competition – DXC'11. In *Proc. of the 22nd international workshop on principles of diagnosis* (pp. 267–278).

Poll, S., Patterson-Hine, A., Camisa, J., Nishikawa, D., Spirkovska, L., Garcia, D., ... Lutz, R. (2007, May). Evaluation, selection, and application of model-based diagnosis tools and approaches. In *AIAA infotech@aerospace 2007 conference and exhibit.*

Puig, V., Quevedo, J., Escobet, T., & Pulido, B. (2005). On the Integration of Fault Detection and Isolation in Model Based Fault Diagnosis. In *Proceedings of the 16th international workshop on principles of diagno-*

sis, dx05 (p. 227-232). Pacific Grove, CA, USA.

Ricks, B., & Mengshoel, O. (2009, September). Methods for probabilistic fault diagnosis: an electrical power system case study. In *Annual conference of the prognostics and health management society (phm09).* San Diego, USA.

Roychoudhury, I. (2009). *Distributed diagnosis of continuous systems: Global diagnosis through local analysis*. Unpublished doctoral dissertation, Vanderbilt University.

Roychoudhury, I., Biswas, G., & Koutsoukos, X. (2010). Distributed diagnosis in uncertain environments using dynamic bayesian networks. In *18th mediterranean conference on control & automation (med),* (pp. 1531–1536).

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1996, March). Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, *4*(2), 105-124.

Sweet, A., Feldman, A., Narasimhan, S., Daigle, M., & Poll, S. (2013, September). Fourth international diagnostic competition – DXC'13. In *Proc. of the 24th international workshop on principles of diagnosis* (pp. 224–229).

Ying, J., Kirubarajan, T., Pattipati, K., & Patterson-Hine, A. (2000, Nov). A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *30*(4), 463-473. doi: 10.1109/5326.897073

BIOGRAPHIES

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis of complex physical systems.

Uncertainty in Steady-State Diagnostics of a Current-Pressure Transducer: How Confident are We in Diagnosing Faults?

Shankar Sankararaman[1], Christopher Teubert[2], and Kai Goebel[3]

[1,2] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov
christopher.a.teubert@nasa.gov

[3] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

Current-Pressure (I/P) transducers are effective pressure regulators that can vary the output pressure depending on the supplied electrical current signal, and are commonly used in pneumatic actuators and valves. Faults in current-pressure transducers have a significant impact on the regulation mechanism, and therefore, it is important to perform diagnosis to identify such faults. However, there are different sources of uncertainty that significantly affect the diagnostics procedure, and therefore, it may not be possible to perform fault diagnosis and prognosis accurately, with complete confidence. These sources of uncertainty include natural variability, sensor errors (gain, bias, noise), model uncertainty, etc. This paper presents a computational methodology to quantify the uncertainty and thereby estimate the confidence in the fault diagnosis of a current-pressure transducer. First, experiments are conducted to study the nominal and off-nominal behavior of the I/P transducer; however, sensor measurements are not fast enough to capture brief transient states that are indicative of wear, and hence, steady-state measurements are directly used for fault diagnosis. Second, the results of these experiments are used to train a Gaussian process model using machine learning principles. Finally, a Bayesian inference methodology is developed to quantify the uncertainty and assess the confidence in fault diagnosis by systematically accounting for the aforementioned sources of uncertainty.

1. INTRODUCTION

Current-Pressure transducers (I/P transducer or IPT) are effective pressure regulators that vary the output pressure depending on the supplied electrical current signal. They operate by throttling a nozzle to create a pressure difference across a diaphragm, which, in turn, controls the throttling of a valve. These are often used for supplying precise pressures to control pneumatic actuators and valves. When such transducers are subjected to wear, it may not be possible to efficiently regulate currents so that desired output pressures may be generated. Therefore, it is necessary to constantly monitor the performance of the transducer using efficient health management techniques and continuously perform diagnosis and prognosis, i.e., detect, isolate, and estimate faults and quantify the remaining useful life of the transducer. Wear detection, estimation, and prediction play a critical role in preventing failure, scheduling maintenance, and improving system utility.

An important challenge in health management is the presence of several sources of uncertainty that affect both diagnosis and prognosis. These sources of uncertainty are present in measurement sensors, system models, and the system inputs. Due to these sources of uncertainty, it becomes necessary to quantify the confidence in the results of diagnosis and prognosis. This can be addressed by estimating the uncertainty in the results of diagnosis (Sankararaman & Mahadevan, 2011, 2013) by rigorously accounting for these sources of uncertainty during health monitoring. While these preliminary methods for uncertainty quantification in diagnosis have been developed from a statistical point of view, it is still necessary to explore the applicability of these methods to different types of practical applications where the impact of uncertainty is extremely significant. While the above statistical methods can efficiently diagnose abrupt faults, wear in practical applications is usually continuous and hence, more challenging from the point of diagnosis and uncertainty quantification.

This paper focuses on applying uncertainty quantification methods to continuous wear estimation in the aforementioned current-pressure transducer. Previous studies at NASA Ames Research Center (Teubert & Daigle, 2013) have ob-

Shankar Sankararaman et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

served that there is a significant amount of uncertainty during the health monitoring of the aforementioned current-pressure transducer; however, the effects of uncertainty on the IPT steady-state diagnosis and prognosis were not studied because simplistic look-up tables had been used for fault estimation. In order to apply rigorous uncertainty quantification methods, it is first necessary to identify and address certain application-specific challenges. In the case of the current-pressure transducer, the challenge lies in obtaining useful information from the sensors used in the health monitoring system. To begin with, there is a significant amount of noise and uncertainty in the sensor measurements. More importantly, the sensors are not fast enough to capture brief transient states; this can either be a result of sensor technological limits, or budgetary constraints on sensor selection (as sensors with higher resolution and higher sampling frequencies are generally more expensive). Many modern wear estimation diagnostic techniques rely on the measurement of the system's transient states (Daigle & Goebel, 2013; Orchard & Vachtsevanos, 2009; Saha & Goebel, 2009; Luo, Pattipati, Qiao, & Chigusa, 2008), and therefore, these techniques cannot be used for diagnosis of the current-pressure transducer. In order to overcome this challenge, researchers at NASA Ames Research Center (Teubert & Daigle, 2013) are pursuing a diagnostic methodology that relies only on steady-state measurements without using any transient information. Therefore, it is necessary to rely on such steady-steady measurements while quantifying the uncertainty in diagnosis.

The primary goal of this paper is to develop a computational methodology to assess the impact of the different sources of uncertainty on wear estimation in the current-pressure transducer, and in turn, quantify the uncertainty in diagnostics. First, experimental data are collected to study the relationship between the input currents, fault magnitudes, and the output pressures, and the resulting data are used to develop a Gaussian process model that can predict the output pressures as a function of input currents and fault magnitude. This model is built offline using principles of machine learning, and then used for diagnosis during online health monitoring. A Bayesian inference-based methodology is developed to quantify the extent of wear, and the associated uncertainty. This analysis is continuously performed in order to continuously estimate the wear and thereby, the fault magnitude can be quantified as a function of time. The Bayesian inference-based methodology provides a systematic framework for including different sources of uncertainty and quantifying the combined effect of the different sources of uncertainty on fault estimation uncertainty, thus providing an estimate in the confidence in diagnosis.

The paper is organized as follows. Section 2 describes the current-pressure transducer in detail, and explains the various modeling and experimental aspects of the transducer. Section 3 describes the Gaussian process modeling methodology

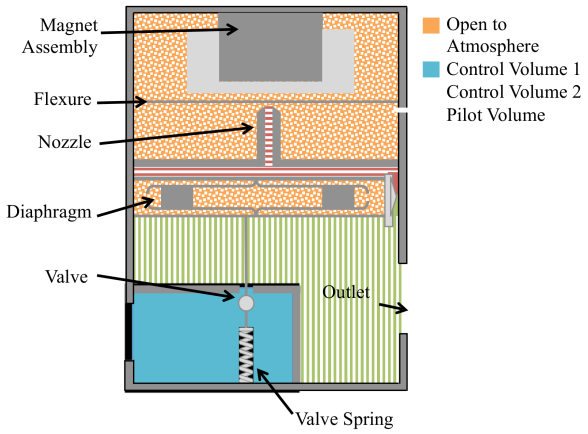


Figure 1. Current/pressure transducer schematic.

Figure 2. Current/pressure transducer.

that is used as a machine learning tool to model the nominal and off-nominal behavior of the current-pressure transducer, and in turn used for diagnosis. Section 4 describes the Bayesian inference-based methodology for quantifying the uncertainty in diagnosis, using the aforementioned Gaussian process model. A simplistic metric for confidence assessment in diagnostics is also presented. Finally, the numerical results are described in Section 5, and conclusions are presented in Section 6.

2. DESCRIPTION OF THE TRANSDUCER

This section describes the behavior of the current-pressure transducer in detail, by exploring both nominal and off-nomoinal (faulty) conditions. Consider a Marsh Bellofram Type 1000 IPT, as shown in Figures 1 and 2. Some specifications for this IPT are included in Table 1 (Marsh Bellofram, n.d.). This particular transducer was chosen because of its use for cryogenic propellant loading applications, and, specifically in the Prognostics Demonstration Testbed at NASA Ames Research Center (Kulkarni, Daigle, & Goebel, 2013).

The IPT is divided into three distinct control volumes (CVs): Control Volume 1 (CV1) at the inlet, Control Volume 2 (CV2)

Table 1. IPT specifications

Name	Type 1000 IPT
Manufacturer	Marsh Bellofram
Supply Pressure Range	18-100 psig
Input Signal Range	4-20 mA
Output Pressure Range	3-15 psig

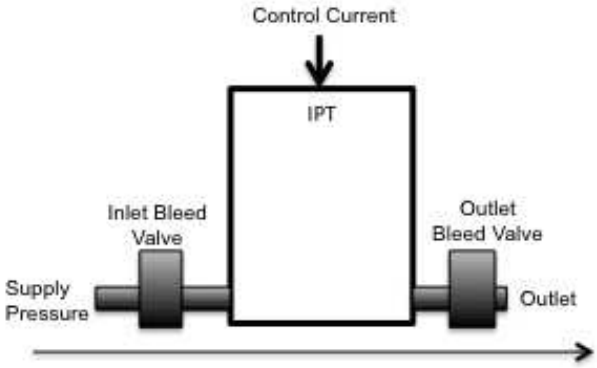


Figure 3. IPT testing configuration

at the outlet, and the Pilot Control Volume (CVP) at the nozzle. Each control volume is marked in a different color and pattern in Figure 1. The IPT output pressure varies with the current supplied to the magnet assembly. When the current is high, the magnet assembly throttles the flow out of the pilot nozzle, allowing less air to escape through the nozzle. With a low input current, more gas escapes from the nozzle thereby lowering the pilot pressure. The pressure difference across the diaphragm moves the valve, which adjusts the gas flow between CV1 and CV2. Adjusting this flow changes the pressure in CV2, and thus provides a direct mechanism to regulate the outlet pressure. In past research efforts, the behavior of this transducer has been modeled using a physics-based approach (Teubert & Daigle, 2013, 2014); however, this model is not used in this paper. Instead, a completely data-driven approach is used for both performance prediction and health monitoring. The experimental set-up for generating data is described in the next subsection.

2.1. Experimental setup

In order to study the nominal and faulty performance of the transducer, a series of experiments were conducted using the Prognostics Demonstration Testbed at NASA Ames Research Center. The Prognostics Demonstration Testbed (Kulkarni et al., 2013) was developed to demonstrate cryogenic refueling valve prognosis. This testbed included an I/P Transducer that was used to operate a large valve. The section of the testbed including the I/P Transducer is illustrated in Figure 3.

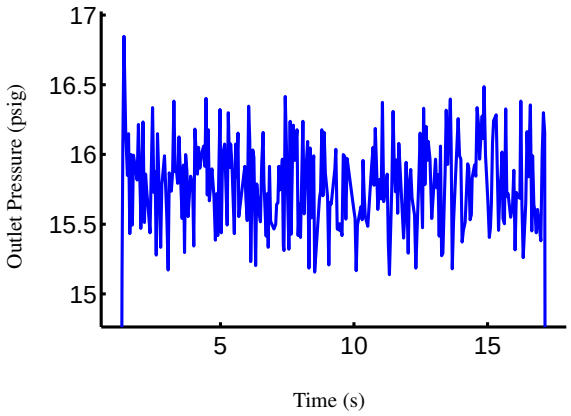


Figure 4. IPT outlet pressure with time

As seen in the figure, two bleed valves were installed on the IPT line: one upstream, and one downstream. These valves were used to simulate inlet and outlet leaks, respectively. A pressure of 75 psig is supplied using a pump. Data were collected from pressure sensors located before the inlet bleed valve and after the outlet bleed valve at a frequency of 16.8 Hz using an 8-slot NI cDAQ-9188 Gigabit Ethernet chassis data acquisition (DAQ) system (Kulkarni et al., 2013). A control input is supplied to the IPT. A separate control input is supplied to the bleed valves to create a leak.

2.2. Nominal IPT Behavior

The IPT documentation indicated the IPT should produce an outlet pressure of 3 and 20 psig when supplied a signal current of 4 and 20 mA, respectively (Marsh Bellofram, n.d.). In this range, the pressure changes linearly with input current.

In practice, IPT behavior is much more difficult to understand. Noise as much as 10% was observed in measurements of outlet pressure, as seen in the experimental data included in Figure 4. This figure shows the measured outlet pressure with time. This noise complicates the process of measuring the steady-state pressure, and thereby complicates the diagnosis procedure. Hence, a rigorous diagnosis methodology should be able to separate the effect of the noise; in fact, this is a prominent feature of the diagnosis method proposed in this paper (in Section 4).

Additionally, it was observed that the pressure at a given input current would vary from day to day but was generally constant over the course of one experiment. We will henceforth refer to this phenomena as "wandering set-point". A histogram showing the spread of steady state pressure measurements over 676 cycles with an input current of $4mA$ is included in Figure 5. In this figure, the input current predicted by the model and documentation is indicated by a dashed red line.

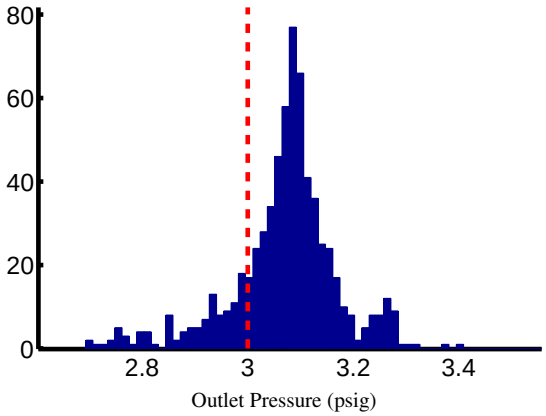


Figure 5. Histogram of IPT steady-state outlet pressure for an input current of 4 mA

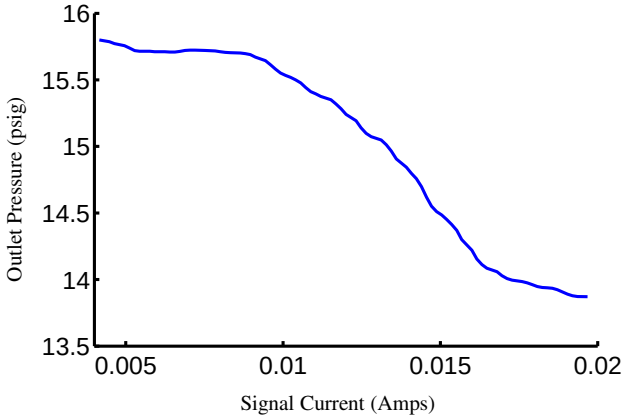


Figure 6. Outlet leak

An experiment was conducted to determine if wandering set-point is observable over the course of one experiment. These experiments found that after 200 minutes of consistent operation there was no observable wandering set-point. From this it was concluded that this phenomena will not occur during the course of a single experiment. In this paper, the wandering set-points are directly included into the data-driven modeling framework, and accounted for during diagnosis, as explained in Sections 4 and 5.

2.3. IPT Wear

Through discussions with the manufacturers and with users of I/P transducers and similar components four possible wear modes were indicated. These wear modes are described below:

1. **Leaks** A leak could occur at the inlet (inlet leak), at the outlet (outlet leak), at the valve (valve seat leak), or at the nozzle (pilot leak).
2. **Spring Weakening** A weakening of the valve spring, the diaphragm, or the flexure. This will decrease the spring coefficient of the effected system.
3. **Valve Impediment** A impediment or "clog" at the valve opening between $CV1$ and $CV2$. This can be caused by foreign object contamination.
4. **Magnet Assembly Weakening** A weakening of the magnet assembly with use.

Though all these faults are possible, this paper focuses only on outlet leak faults. Outlet leaks were chosen because they are well understood and can be directly simulated while performing experiments. Only this fault and inlet leaks can be simulated in the laboratory with our current experimental setup. Studies show that introducing an inlet leak has very little effect on IPT performance (Teubert & Daigle, 2013). Other faults will be considered in future work.

A bleed valve to the atmosphere was introduced into the experimental setup after the IPT to simulate outlet leaks. Each bleed valve simulates a leak up to 3/64" in diameter. IPT Performance with various levels of outlet leaks can be seen in Figure 6.

As mentioned in the previous subsection on nominal behavior, between experiments the IPT behavior will change slightly in the "wandering set-point" phenomenon. This phenomenon also affects IPT wear behavior, and will be accounted for during modeling in Section 3, and during diagnosis in Section 4.

3. GAUSSIAN PROCESS MODELING

The experimental data used to study the performance of the current-pressure transducer is then used to train a Gaussian process data-driven model. This model predicts the outlet pressure as a function of input current, fault magnitude (outlet leak fault), and the wandering set-points. The gaussian process model is a powerful multi-dimensional interpolation technique based on spatial statistics. It is increasingly being used to build surrogates to replace expensive computer simulations in order to facilitate efficient optimization and uncertainty quantification (Rasmussen, 2004; Santner, Williams, & Notz, 2003). The GP model is preferred in this research for the following reasons: (1) it is not constrained by functional forms; (2) it is capable of representing highly nonlinear relationships in multiple dimensions; and (3) can estimate the prediction uncertainty which depends on the number and location of training data points.

The basic idea of the GP model is that the response values Y evaluated at different values of the input variables $\boldsymbol{X}$, are modeled as a Gaussian random field, with a mean and covariance function. Suppose that there are m training points, x_1, x_2, x_3 ... x_m of a d-dimensional input variable vector

($d = 4$ in this paper), yielding the output values $Y(x_1)$, $Y(x_2)$, $Y(x_3)$ ... $Y(x_m)$. The training points can be compactly written as x_T vs. y_T where the former is a $m \times d$ matrix and the latter is a $m \times 1$ vector. Suppose that it is desired to predict the response (output values y_P) corresponding to the input x_P, where x_P is $n \times d$ matrix; in other words, it is desired to predict the output at n input combinations simultaneously. Then, the joint density of the output values y_P can be calculated as:

$$p(y_P|x_P, x_T, y_T; \Theta) \sim N(m, S) \tag{1}$$

where Θ refers to the hyperparameters of the Gaussian process, which needs to be estimated based on the training data. The prediction mean and covariance matrix (m and S respectively) can be calculated as:

$$\begin{aligned} m &= K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} y_T \\ S &= K_{PP} - K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} K_{TP} \end{aligned} \tag{2}$$

In Eq. 2, K_{TT} is the covariance function matrix (size $m \times m$) amongst the input training points (x_T), and K_{PT} is the covariance function matrix (size $p \times m$) between the input prediction point (x_P) and the input training points (x_T). These covariance matrices are composed of squared exponential terms, where each element of the matrix is computed as:

$$K_{ij} = K(x_i, x_j; \Theta) = -\frac{\theta}{2}[\sum_{q=1}^{d} \frac{(x_{i,q} - x_{j,q})^2}{l_q}] \tag{3}$$

Note that the above computations require the estimate of the multiplicative term (θ), the length scale in all dimensions (l_q, $q = 1$ to d), and the noise standard deviation (σ_n). These constitute these hyperparameters ($\Theta = \{\theta, l_1, l_2 \ ... \ l_d, \sigma_n\}$). These hyperparameters are estimated based on the training data by maximizing the following log-likelihood function:

$$\begin{aligned} log\ p(y_T|x_T; \Theta) = &-\frac{y_T^T}{2}(K_{TT} + \sigma_n^2 I)^{-1} y_T \\ &- \frac{1}{2} log|(K_{TT} + \sigma_n^2 I)| + \frac{d}{2} log(2\pi) \end{aligned} \tag{4}$$

Once the hyperparameters are estimated, the Gaussian process model can be used for predictions using Eq. 2. Note that the "hyperparameters" of the Gaussian process are different from the "parameters" of a generic parametric model (for e.g. linear regression model). This is because, in a generic parametric model, it is possible to make predictions using only the parameters. For the Gaussian process model, all the training points and the hyperparameters are both necessary to make predictions, even though the hyperparameters may have estimated previously. For details of this method, refer to (Rasmussen, 2004; Chiles & Delfiner, 1999).

Once the training points are selected and the Gaussian process model is constructed, it can be used for diagnosis and quantifying the uncertainty in diagnosis, as explained in Section 4.

4. WEAR ESTIMATION AND UNCERTAINTY QUANTIFICATION

Wear estimation is the process of estimating the current extent of wear (i.e., quantifying the fault magnitude) on a system. This is important for prognostics (predicting failure and remaining useful life), scheduling maintenance, and triggering automated mitigation actions. This is often done using methods such as a Kalman Filter or Particle Filter (Arulampalam, Maskell, Gordon, & Clapp, 2002; Daigle, Saha, & Goebel, 2013). In this paper, recall that only steady-state measurements have been used and the transients are completely ignored. For this reason, tracking is not applicable and filtering approaches will not be suitable for wear estimation. Therefore, it is necessary to develop an algorithm that can estimate the extent of wear. Previously (Teubert & Daigle, 2013), a lookup table method was used for fault estimation. This method was chosen because of its fast, efficient nature and its ability to be applied to both linear and non-linear systems. However, this method can neither systematically account for the different sources of uncertainty nor quantifying the uncertainty in fault estimation. Hence, this paper uses the previously described Gaussian process model and Bayesian inference to quantify uncertainty in fault estimation.

As mentioned previously, this paper focuses on the outlet leak. This fault has a definite and measurable effect on the outlet pressure and can be simulated in the lab. As the leak grows in size, more gas escapes through the outlet. For a leak of 5 mm^2, the outlet pressure decreases by 2.101 psig for a high signal current and by 0.207 psig for a low signal current.

This paper focuses on quantifying the amount of wear by approaching fault estimation as a parameter estimation problem. In this technique, input-output measurements (obtained from the health monitoring sensors) are directly used to estimate the magnitude of fault; the input corresponds to the signal current (denoted by I) to the IPT, the output corresponds to the outlet pressure (denoted by P), and the magnitude of fault (wear) is denoted by θ. Further, the outlet pressure also depends on the two set-points (denoted by α_1 and α_2) that are measured during the course of health monitoring. The entire procedure for fault estimation and uncertainty quantification is described through the stepwise procedure, as shown in flowchart in Fig. 7. Each of these steps are explained in detail below.

4.1. Offline: Gaussian Process Model Development

Any parameter estimation technique relies on the existence of a forward model that can compute the quantity being measured as a function of the fault magnitude. This forward

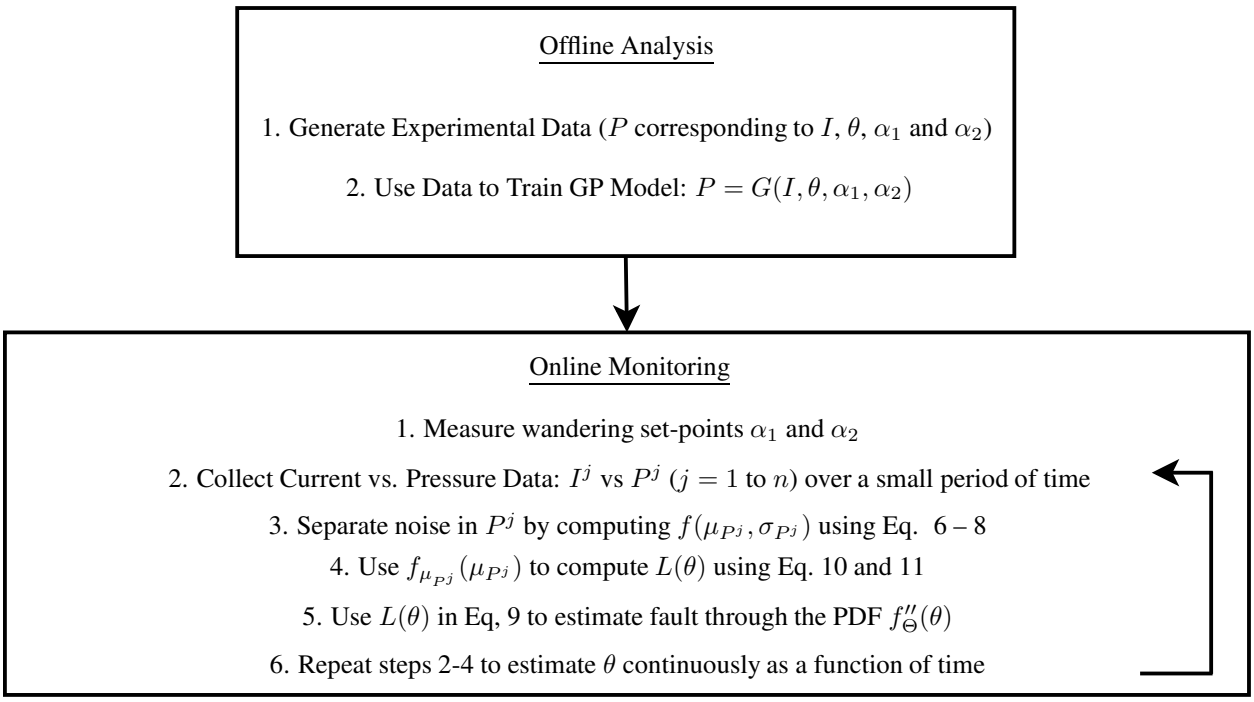


Figure 7. Stepwise Diagnosis Procedure

model is represented as:

$$P = G(I, \theta, \alpha_1, \alpha_2) \tag{5}$$

The forward model can either be physics-based or data-driven. In this paper, a fully data-driven approach is pursued. Experimental data are used to train the Gaussian process model as described in Section 3. While a rigorous design of experiments is not performed (due to the challenges involved in the experimental set up and data collection), six different runs are used to generate the training data. Each experimental run corresponds to a single pair of set-points. Within each experimental run, the fault magnitude increases gradually (as shown in Fig 15); for each value of fault magnitude, two values of I and the corresponding values of P are measured. All this data are used to train the Gaussian process model offline. After training, the model can be used for online diagnosis.

4.2. Online: Measurements and Set-Points

For performing diagnosis, the first step is to measure to set-points (α_1 and α_2); As mentioned in Section 2 IPT behavior can change over time (the "Wandering Setpoint Phenomenon"). The set-points are the outlet pressure of the undamaged system given a control input of 4 and 20 mA (the operational extremes). These values are used to quantize the wandering setpoint magnitude. Wear behavior is then dependent on the values of these set points.

Then, a small time period within which the fault magnitude is likely to be constant is considered; the current values and corresponding outlet pressure values are measured during this time period. Let I^j and P^j ($j = 1$ to n) denote the measured input-output data. The goal is to use these measurements to estimate the magnitude of fault accounting for the noise in the measurement data and other sources of uncertainty. This is accomplished through the use of the above constructed surrogate model and Bayesian inference (Sankararaman & Mahadevan, 2013). The first step is to explicitly quantify the amount of noise in the data, so that the actual steady state value may be calculated.

4.3. Separating Noise from Steady State Pressure

Consider the input-output data, described in terms of I^j versus P^j ($j = 1$ to n). In the experimental setup, the input current is treated as the independent quantity and can be controlled fully, i.e., it is assumed that there is no uncertainty regarding the current values. However, the P^j corresponds to the steady state pressure that is measured. Typically, this steady state pressure is contaminated with noise. It is important to separate out the effect of such noise. A typical steady state pressure consisting of 252 measurements is shown in Fig. 8.

One way to quantify the actual steady state value is to simply compute the average of all the measurements; however, this is not an effective treatment of uncertainty. Therefore,

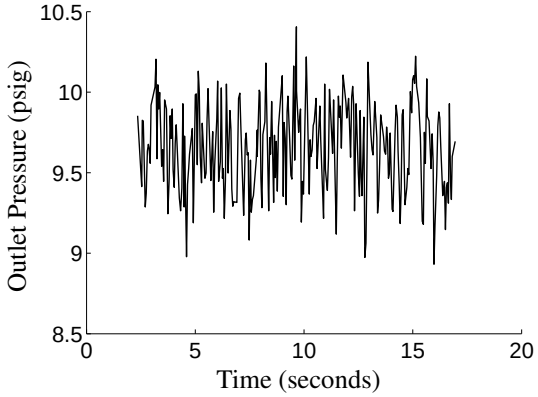


Figure 8. Steady state outlet pressure values

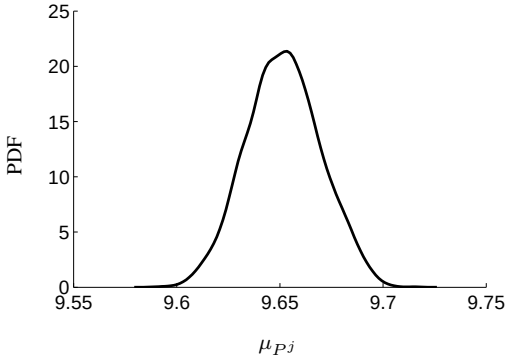


Figure 9. PDF of steady-state value (μ_{Pj})

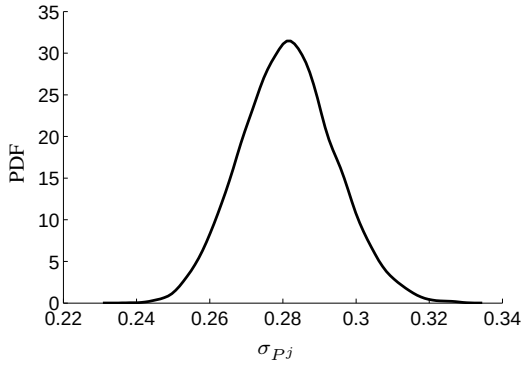


Figure 10. PDF of the standard deviation of measurement error (σ_{Pj})

this paper develops a new method to individually quantify the constant value and the noise magnitude. To this end, consider the separation of the steady state value into the constant term and noise as:

$$P^j = \mu_{Pj} + \epsilon_{Pj} \tag{6}$$

where μ_{Pj} is the actual constant steady-state value and ϵ_{Pj} is the measurement error. Further, it is assumed that the measurement error ϵ_{Pj} follows a Gaussian distribution with zero mean and standard deviation equal to σ_{Pj}. Then, based on all the measurements in Fig. 8, Bayes theorem can be used to estimate the probability distributions of both μ_{Pj} and σ_{Pj}. If the N_j (equal to 252 in Fig. 8) measurements are denoted as P_k^j ($k = 1$ to 252), then, the likelihood function $L(\mu_{Pj}, \sigma_{Pj})$ is constructed as:

$$L(\mu_{Pj}, \sigma_{Pj}) \propto \prod_{k=1}^{k=N_j} \frac{1}{\sqrt{(2\pi)}\sigma} exp\left(- \left[\frac{(\mu_{Pj} - P_k^j)^2}{\sigma_{Pj}^2} \right] \right) \tag{7}$$

Then, this likelihood function is used to estimate the joint PDF of μ_{Pj} and σ_{Pj} using Bayes theorem, as:

$$f(\mu_{Pj}, \sigma_{Pj}) = \frac{L(\mu_{Pj}, \sigma_{Pj})}{\int L(\mu_{Pj}, \sigma_{Pj}) d\mu_{Pj} d\sigma_{Pj}}. \tag{8}$$

Note that the above equation is simply a variation of Bayes' theorem; the prior distribution has been canceled in both the numerator and the denominator (inherently assuming that a constant prior has been used). It is not necessary to evaluate the above integral explicitly; instead, slice sampling (Neal, 2003) is used to directly estimate samples of μ_{Pj} and σ_{Pj} from the posterior distribution on the right hand side of the above equation. For the steady state in Fig. 8, the PDFs of μ_{Pj} and σ_{Pj} are shown in Fig. 9 and Fig. 10.

4.4. Fault Estimation through Bayesian Inference

Having the steady state, this information along with the GP model can be used to quantify the fault magnitude and the associated uncertainty. In order to achieve this goal, let $f'_\Theta(\theta)$ denote the prior probability distribution of the fault magnitude before collecting measurements; a uniform probability distribution over the entire range of possible fault magnitudes is assumed in this paper. Then, using the available input-output data, the posterior distribution of the fault magnitude (denoted by $f''_\Theta(\theta)$) is computed as:

$$f''_\Theta(\theta) = \frac{f'_\Theta(\theta) L(\theta)}{\int f''_\Theta(\theta) L(\theta) d\theta} \tag{9}$$

where $L(\theta)$ is the likelihood function of θ, defined as being proportional to the probability of observing the given input-output data conditioned on the value of the fault magnitude θ. The likelihood function, i.e., $L(\theta)$ is constructed using the estimated steady state pressure value. Recall that μ_{Pj} denotes the constant steady state pressure value and $f_{\mu_{Pj}}(\mu_{Pj})$ denotes the corresponding PDF.

Then, the likelihood function for the i^{th} input-output datapoint is expressed as:

$$L(\theta^i) \propto f_{\mu_{Pj}}(\mu_{Pj} = G(I^j, \theta, \alpha_1, \alpha_2)) \tag{10}$$

Since the n measurements are independent of one another, the combined likelihood can be calculated as:

$$L(\theta) = \prod_{i=1}^{i=n} L(\theta^i) \tag{11}$$

Then, this likelihood function is substituted into Eq. 9, and the posterior PDF of the fault magnitude θ is computed. While direct integration (Sankararaman, Ling, & Mahadevan, 2010) is used in this paper, advanced MCMC sampling methods such as slice sampling (Neal, 2003) can also be used. This procedure is repeated continuously to estimate the PDF of the fault magnitude as a function of time.

4.5. Metric for Assessing Confidence in Diagnostics

A common practice in health management is to not use the entire PDF information and simply use some central tendency of the above calculated PDF (say, mean, median, or mode) as the final diagnostic estimate. However, this procedure loses information regarding uncertainty and can lead to erroneous results. That is why it is important to quantify the confidence is diagnostic assessments. This paper discusses a simple confidence metric to address this issue.

For example, consider the mode of the PDF $f''_\Theta(\theta)$. By definition, the mode of a probability distribution has the highest likelihood of occurrence and hence is the most likely value. Therefore, the mode of the PDF $f''_\Theta(\theta)$ would be the most likely fault magnitude value. However, this implies that the true fault value may have a smaller likelihood of occurrence. Therefore, a simple way to compute a confidence metric would be to assess how far the mode (denoted by θ_C) is probabilistically away from the true estimate (denoted by θ_T). This can be computed mathematically using the likelihood ratio:

$$M = \frac{f''_\Theta(\theta_T)}{f''_\Theta(\theta_C)} \tag{12}$$

This ratio will be equal to one when the estimated mode value coincides with the true value, and in all other cases, the metric will be less than equal to one. The metric provides a probabilistic measure of confidence in the estimated fault by comparing its likelihood against the true fault magnitude. For practical purposes, the above metric can also be expressed in terms of percentage, as illustrated later in this paper.

5. NUMERICAL RESULTS

This section presents the numerical results of diagnosis uncertainty quantification on a current-pressure transducer.

5.1. Training the Gaussian Process Model

The first step is to use the experimental data to train the Gaussian process model. This model has four input quantities:

1. Fault magnitude
2. Current magnitude
3. Set-point I
4. Set-point II

For every combination of the above four quantities, the outlet steady state pressure needs to computed by the gaussian process model ($P = G(I, \theta, \alpha_1, \alpha_2)$). Hence, experimental data that depicts the variation of output pressure with respect to the four input quantities are collected and used to train the GP model. There are seven sets of data, and each set corresponds to one value of set-point I and set-point II. These experimental are shown in Figures 11—14

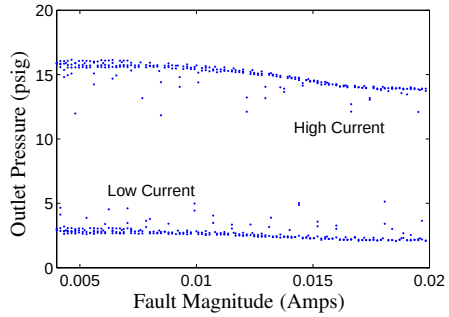


Figure 11. Fault magnitude vs. output pressure

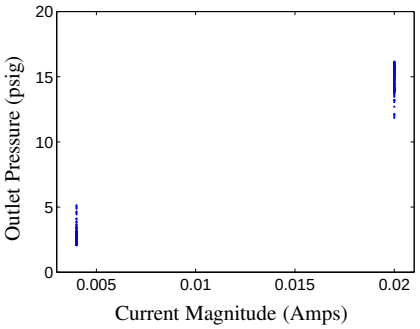


Figure 12. Current magnitude vs. output pressure

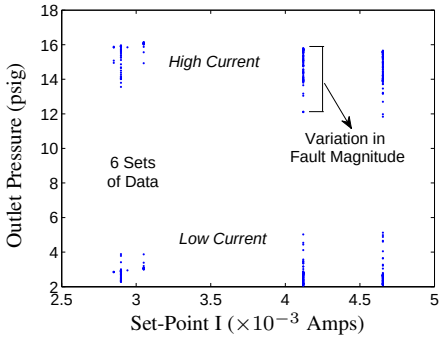


Figure 13. Set-Point I vs. output pressure

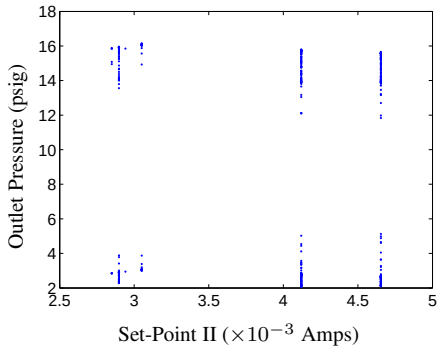


Figure 14. Set-Point II vs. output pressure

All of the above information is used to train the Gaussian process model using the procedure in Section 3. This model is used for diagnosis and quantifying the uncertainty in diagnosis.

5.2. Diagnosis: Numerical Illustration

Consider a set of current versus (steady state) outlet pressure measurements that are available through health monitoring, as shown in Fig. 15. Note that the Gaussian process model is useful for forward evaluation, i.e., to compute the outlet pressure as a function of fault magnitude and input current, and this model needs to be evaluated for multiple values of fault magnitude in order to estimate the correct fault magnitude.

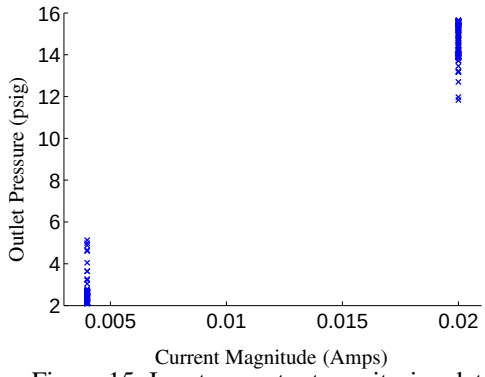


Figure 15. Input vs. output monitoring data

Two values of current are applied in an alternating manner: First, a current of 0.004 amps, and then a current of 0.02 amps. The fault magnitude is assumed to be constant over this time window. This procedure is repeated as the fault magnitude increases over time. The set-points for the above monitoring data are found to be equal to 4.65 and 15.58 milliamps. Using the Gaussian process model, and the Bayesian inference methodology explained earlier in Section 4, the fault magnitude is estimated continuously as a function of time. To estimate the fault magnitude, one low value of current and one high value of current, and the corresponding outlet pressures are considered. Since 198 sets of measurement are available and every two correspond to a single value of fault magnitude, Bayesian inference is applied 98 times to quantify the fault magnitude.

An arbitrary set of current-pressure values is chosen for the purpose of illustration; the outlet pressure values given signal currents of 0.004 amps and 0.02 amps are equal to 2.68 Pa and 15.56 Pa respectively. For these set of values, the fault magnitude is estimated using Bayesian inference; the estimated PDF and the true value are indicated in Fig. 16. Note that the mode does not correspond to the true fault magnitude.

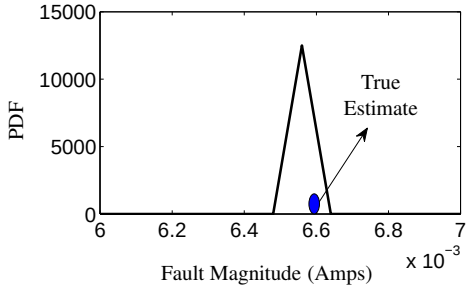


Figure 16. PDF of fault magnitude

Such computation is continuously performed with time, and the mode of the distribution is plotted against the true fault magnitude value, as shown in Fig. 17. While absolute time is not meaningful, Fig. 17 shows the number of the instance (1 through 99) in which diagnosis is performed. It can be seen that the mode approximately matches well the true fault magnitude (since the fault magnitude varies over a range, it is not possible to see succinct differences between the mode and true fault magnitudes). The methodology consistently estimates the fault magnitude and the true fault magnitude is contained within reasonable bounds of the predicted uncertainty.

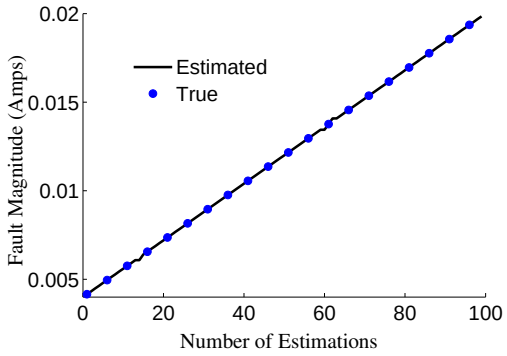


Figure 17. Fault magnitude: estimated (mode) vs. true

In addition to the mode of the fault estimate, the standard deviation is also plotted in Fig. 18, similar to Fig. 17. Note

that the standard deviation is small, as seen from Fig. 16 and Fig. 18.

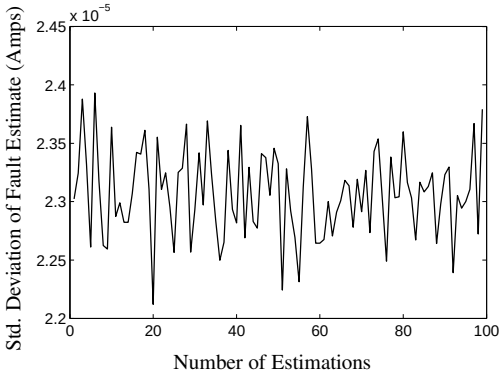


Figure 18. Uncertainty in diagnosis

However, using the proposed statistical methods, it is possible to quantify the extent of agreement between the estimated fault and true magnitude, thereby quantifying the amount of confidence in diagnosis. The metric proposed earlier in Section 4.5 (ratio of PDFs measured at the mode and the true value) is quantified and plotted (as percentage) in Fig. 19.

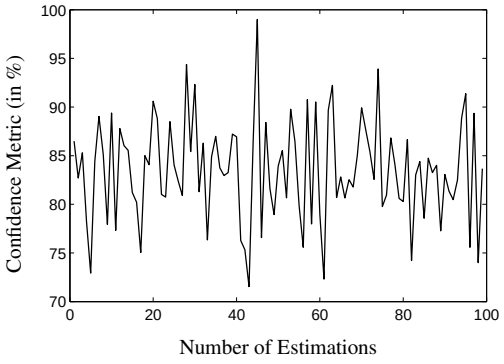


Figure 19. Confidence in diagnosis

As seen from Fig. 19, it is seen that the confidence metric is always less than 100%, suggesting that it is practically impossible to precisely estimate the true fault magnitude. A rigorous treatment of uncertainty addresses this issue by estimating the entire PDF of the fault magnitude instead of using any central tendency such as the mean, median, mode, etc.

6. CONCLUSION

This paper proposed a data-driven methodology for fault estimation and uncertainty quantification in the steady-state diagnosis of a current-pressure transducer (IPT). Such transducers are efficient electromechanical devices that can be used to control the output pressure depending on the signal current. When faults are present in these transducers, the desired pressure output may not be obtained. Therefore, it is necessary to monitor to performance of these transducers, detect the presence of faults and estimate the fault magnitude.

This is a significant challenge in diagnosis due to several sources of uncertainties associated with monitoring the heath of the transducer. To begin with, the sensors used to monitor the performance may be affected by sensor noise. Further, it may not be precisely possible to predict the performance of the transducer and this may add further uncertainty; therefore it becomes necessary to quantify the confidence in fault diagnosis.

A Bayesian inference-based methodology was used for uncertainty quantification in diagnostics, and the amount of wear (fault) was quantified as a function of time. This approach can not only systematically account for the various sources of uncertainty in the health monitoring but also quantify the uncertainty in the fault estimate, resulting in a measure of confidence in diagnosis. Experimental data were collected offline and used to develop a Gaussian process model that can predict the outlet pressure as a function of fault magnitude and input current. This Gaussian process model was then used in online diagnosis; the probability distribution of the fault magnitude and the confidence in diagnostics was estimated.

Numerical results show considerable promise of the proposed methodology. Future work may include considering multiple, simultaneous fault modes where it is necessary to quantify the uncertainty in both fault isolation and fault estimation. It is also necessary to study the effect of diagnostic uncertainty on prognosis, by quantifying the uncertainty in the remaining useful life of the transducer.

ACKNOWLEDGMENT

The study reported in this paper was partly funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD) and by the AGSM (Advanced Ground Systems Maintenance) project under the Ground Systems Development and Operations Program in the Human Exploration and Operations Mission Directorate. The authors also thank Dr. Chetan Kulkarni and George Gorospe at NASA Ames Research Center for their help with preparing and conducting IPT experiments on the Pneumatic Valve Testbed.

REFERENCES

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Chiles, J., & Delfiner, P. (1999). *Geostatistics: modeling spatial uncertainty* (Vol. 344). Wiley-Interscience.

Daigle, M., & Goebel, K. (2013, May). Model-based prog-

nostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 535-546.

Daigle, M., Saha, B., & Goebel, K. (2013, March). A comparison of filter-based approaches for model-based prognostics. In *Proceedings of the IEEE aerospace conference.*

Kulkarni, C., Daigle, M., & Goebel, K. (2013, sep). Implementation of prognostic methodologies to cryogenic propellant loading testbed. In *IEEE AUTOTESTCON 2013.*

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Marsh Bellofram. (n.d.). Type 1000 i/p & e/p transducers [Computer software manual].

Neal, R. M. (2003). Slice sampling. *Annals of statistics*, 705–741.

Orchard, M., & Vachtsevanos, G. (2009, June). A particle filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*(3-4), 221-246.

Rasmussen, C. (2004). Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, 63–71.

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society.*

Sankararaman, S., Ling, Y., & Mahadevan, S. (2010). Statistical inference of equivalent initial flaw size with complicated structural geometry and multi-axial variable amplitude loading. *International Journal of Fatigue*, *32*(10), 1689–1700.

Sankararaman, S., & Mahadevan, S. (2011). Uncertainty quantification in structural damage diagnosis. *Structural Control and Health Monitoring*, *18*(8), 807–824.

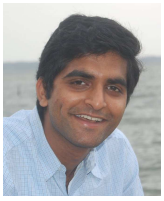
Sankararaman, S., & Mahadevan, S. (2013). Bayesian methodology for diagnosis uncertainty quantification and health monitoring. *Structural Control and Health Monitoring*, *20*(1), 88–106.

Santner, T., Williams, B., & Notz, W. (2003). *The design and analysis of computer experiments*. New York: Springer-Verlag.

Teubert, C., & Daigle, M. (2013, October). I/p transducer application of model-based wear detection and estimation using steady state conditions. In *Proceedings of the annual conference of the prognostics and health management society* (p. 134-140).

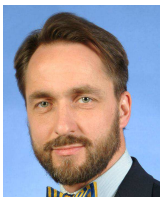
Teubert, C., & Daigle, M. (2014, March). Current/pressure transducer application of model-based prognostics using steady state conditions. In *Proceedings of the IEEE aerospace conference.*

BIOGRAPHIES



Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the American Institute of Aeronautics (AIAA), the American Society of Civil Engineers (ASCE), the Institute for Electrical and Electronics Engineers (IEEE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.

Christopher Teubert received his B.S. in Aerospace Engineering from Iowa State University in 2012. While at Iowa State University, he conducted research on asteroid deflection mission design and asteroid fragment propagation for Iowa State Universitys Asteroid Deflection Research Center (ADRC). Previous to his current position he worked as a spacecraft systems engineer for a Mars sample return mission as part of the NASA Academy Program. He is currently developing software tools and researching algorithms for diagnostics, prognostics, and system health management for Stinger Ghaffarian Technologies, Inc. at NASA Ames Research Center in the Prognostic Center of Excellence (PCoE).



Kai Goebel is the Area Lead for Discovery and Systems Health at NASA Ames where he was director of the Prognostics Center of Excellence during the time this research was conducted. After receiving the Ph.D. from the University of California at Berkeley in 1996, Dr. Goebel worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist before joining NASA. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion and his interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 17 patents and has published more than 250 papers in the area of systems health management.

Equipment Health Monitoring with Non-Parametric Statistics for Online Early Detection and Scoring of Degradation

Maizura Mokhtar[1], Joseph C. Edge[2] and Andrew R. Mills[1]

[1] *Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, United Kingdom*
{*m.mokhtar, a.r.mills*}*@sheffield.ac.uk*

[2] *Rolls-Royce plc, PO Box 31, Derby, England*
joseph.edge@rolls-royce.com

ABSTRACT

This paper develops a health monitoring scheme to detect and trend degradation in dynamic systems that are characterised by multiple parameter time-series data. The presented scheme provides early detection of degradation and ability to score its significance in order to inform maintenance planning and consequently reduce disruption. Non-parametric statistics are proposed to provide this early detection and scoring. The non-parametric statistics approximate the data distribution for a sliding time window, with the change in distribution is indicated using the two-sample Kolmogorov-Smirnov test. Trending the changes to the signal distribution is shown to provide diagnostic capabilities, with deviations indicating the precursors to failure. The paper applies the equipment health monitoring scheme to address the growing concerns for future gas turbine fuel metering valve availability. The fuel metering unit within a gas turbine is a complex electro-mechanical system, failures of which can be a major source of airline disruption. The application is performed on data acquired from a series of industrial tests performed on large civil aero-engine fuel metering units subjected to varying levels of contaminant. The data exhibits characteristics of degradation, which are identified and trended by the equipment health monitoring scheme presented in this paper.

1. INTRODUCTION

The assessment and trending of novelty within the measured parameters of a dynamic system may be used to diagnose and predict the performance and health of a system, and thus inform activities to reduce the impact of decreasing functional performance. The use of novelty as a measure of health has advantages in that the exact nature of fault characteristics are not required in advance, only a measure of departure from nominal conditions. To generate actionable information, signals are typically processed from raw measurements into a reduced dimension novelty summary value that may be more easily transferred to where it can be trended and interpreted by an asset manager. In line with the aspirations of the novelty detection and trending paradigm to determine any departure from nominal conditions, the novelty assessment scheme should be sensitive to all changes in the underlying system, not only deviations in particular characteristics of signals. A multi-variate equipment health monitoring (EHM) scheme is developed to address these novelty trending objectives.

Early warning of degradation is provided by a novelty scoring metric, which aims to detect the changes in the system dynamic response as the results of the degradation and to trend the degradation significance and severity. The changes in the dynamic response are visible when analysing the measured data distributions. For the work presented in this paper, novelty is defined as the change in the measured signal distribution when compared to a reference distribution, generated from a previous known condition or from its earlier behaviour. The principle of our novelty detection scheme is supported by Andrade et al. (2001) which states: "data derived from measurements taken from an undamaged system will have a distribution with an associated mean and variance; if the system is damaged, then, there may be a change to its mean, variance, or both". Online indication and trending of the distribution change, with any order of statistical moments (Scheffer & Heyns, 2001), (Salgado & Alonso, 2006), enable the indication of the system health condition.

Because of this, the proposed EHM scheme does not require an explicit model of normality to be constructed as part of the design and development process. This is in contrast to the work published in (Sohn et al., 2001) and other similar works, (Andrade et al., 2001) (Hall & Mba, 2004), (Kar & Mohanty, 2006), (Subramaniam et al., 2006) and (Zhan & Mechefske, 2007). These papers compare the measured dy-

Maizura Mokhtar et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

namic response against the model of normal(s) and/or fault conditions - therefore, causing a disadvantage because of the requirement for prior knowledge. Our work also enables the early detections of the onset of change in dynamic response, which is also indicative of the degradation.

The novelty scoring is achieved using online non-parametric statistics that approximates the data distribution for the time window consisting of the N number of samples at current time t and compare to the previous N number of samples separated by an interval of S samples. A non-parametric statistical approach is proposed so that this scheme will not be reliant on the prior training of normal and faults conditions. This is a major criteria for the development of the online and unsupervised EHM scheme, because, as indicated in (Modenesi & Braga, 2009), novelty detection is concerned with the identification of unexpected events or regime changes to the system that is not well understood - "The vagueness of the description is inherent to the novelty detection problem, in fact, it is the very centre of the problem: how to detect data whose only particular characteristic is that it has not appeared before?". Furthermore, when variations occur, the variations may cause the need to redesign and reconstruct any system models developed; the development process itself is time consuming, and may not reflect all normal or fault conditions (Zhan & Mechefske, 2007).

By using an online non-parametric statistics (Subramaniam et al., 2006), the approximation of the data distribution adapts over time. The characteristics of the distribution will differ when the conditions of the system have changed, thus changes that are resultant of degradations are identified. Novelty, defined by this work, is the identification of the changes to the distribution, which signifies when a change in the system's conditions have occurred, i.e. the measured dynamic response that is the outcome of degradations. Authors of (Marsland, 2003) and (Modenesi & Braga, 2009) also indicated that novel data or outliers have a large effect on the analysis of the system, which can result in the change to the measured data distribution.

One mechanism to monitor the distribution change is by trending the change in the distributions mean, standard deviation and other statistical moments (e.g. skewness or kurtosis). These summary statistics are not guaranteed to unambiguously measure all the different changes that may occur in the data. In addition, as the number of variables in the analyses increases, the co-relations between parameters should also be calculated, and thus the number of calculations increases non-linearly ($O(n^2)$). Modern complex systems have a combination of multiple sensed parameters that all may contribute to the efficacy of monitoring (Subramaniam et al., 2006). Therefore, an alternative generic measure of distribution change is advantageous and is proposed in this paper.

We apply this scheme to a component previously identified as a source of high disruption and service cost to aero-engine manufacturers (Eleffendi et al., 2012). The fuel metering unit (FMU) within a gas turbine is a complex electro-mechanical system. Failures to the FMU can be a major source of airline disruption. The system operates in a harsh environment where high temperatures and fuel impurities can lead to system degradation and functional failure. Fuel impurities, often categorised as contaminants, are one of the culprits that cause system degradation. Contaminants accumulate in fuel system filters, nozzles, the walls of control valves and other sliding components. These accumulations resulted in increased friction, which can, in addition to other failure mechanisms, result in valve seizure and in-flight shutdown. Early detection of this degradation can inform maintenance planning and avoid in-service events, which helps minimise disruptions.

The paper presents the multivariate EHM scheme that performs early diagnosis and trending of the FMU degradation as a result of friction increase. The EHM scheme uses non-parametric statistics. The non-parametric status is discussed in Section 2. Section 3 describes the FMU used to test and analyse the capabilities of the EHM scheme and Section 4 discusses the results produced. Section 5 concludes the paper.

2. NON-PARAMETRIC STATISTICS FOR NOVELTY DETECTION AND SCORING

The novelty detection scheme proposed is performed by comparing the differences between the two distributions: the current distribution and the previous distribution measured. If the system is in nominal conditions and at non-transient operations, the change should be minimal. If the system performance degrades, a change in the distribution between current and previous is indicated. Changes in the distribution are indicated using a multivariate two-sample Kolmogorov-Smirnov test. The Kolmogorov-Smirnov test signifies the probability whether the two underlying probability distributions differs. The test compares two empirical cumulative distribution functions (ECDFs) and for the work presented in this paper, the two ECDFs are the current and previous distributions. This enables trending of any system change.

2.1. Multivariate Two-sample Kolmogorov-Smirnov Test

Since different data sets, or different distribution functions, have differing cumulative density functions, one can establish the likelihood that two sets of data are originating from the same distribution function by measuring the differences between their ECDFs. The ECDF for the N samples of variable v is defined by Eq. (1), and provides a measure of the relative number of samples for v, $v = \{u_1, u_2, ..., u_N\}$, less than or equal to x. $\mathbf{1}\{u_i \leq x\}$ is the indicator of such an event.

$$\mathrm{ECDF}_v(x) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{u_i \leq x\} \quad (1)$$

The two-sample Kolmogorov-Smirnov test compares the two ECDFs by calculating the statistical distance D between the two distributions. The statistical distance D is given by Eq. (2), where $F(x)$ and $R(x)$ are the samples from the ECDFs of $F(\mathrm{x}_1)$ and $R(\mathrm{x}_2)$ respectively (Andrade et al., 2001).

$$D = \max_{-\infty<x<\infty}|F(x) - R(x)| \quad (2)$$

The statistical distance D is converted into a similarity probability using the Kolmogorov-Smirnov p value, defined by Eqs. (3)–(4) (Greenwell & Finch, 2004) (Kar & Mohanty, 2006). The p value provides the metric for novelty scoring.

$$p = Q_{KS}(z) = 2\sum_{j=1}^{\infty}(-1)^{j-1}\exp(-2j^2z^2) \quad (3)$$

$$z = D\sqrt{\frac{N_1N_2}{N_1+N_2}} \quad (4)$$

N_1 is the number of points in $F(\mathrm{x}_1)$ and N_2 is the number of points in $R(\mathrm{x}_2)$. Equation (3) is for when N_1 and N_2 tends to infinity (Kar & Mohanty, 2006).

p-value is a monotonic function with limiting values of:

$$p = Q_{KS}(z) = \begin{cases} 1 & \text{if } z \to 0 \\ 0 & \text{if } z \to \infty \end{cases} \quad (5)$$

If the two distributions are statistically similar (similar ECDFs), Q_{KS} tends towards 1. If the distributions are different, i.e. varied, Q_{KS} will go towards 0. A variation between the two distributions indicates that a novelty has occurred.

In the work presented in this paper, the $F(\mathrm{x}_1)$ and $R(\mathrm{x}_2)$ are the product of the single variate ECDF_v in the multivariate data, calculated using Eq. (6).

$$F(x) = \prod_{v=1}^{V}\mathrm{ECDF}_v(x) \quad (6)$$

where V is the number of variables considered.

Novelty is indicated when $p < 0.90$. $p < 0.90$ is chosen because, based on the critical value approximation which indicates:

$$H = \begin{cases} 0 & \text{if } D < D_{\mathrm{critical}} \\ 1 & \text{if otherwise} \end{cases} \quad (7)$$

$H = 0$ when the two distributions are the same and $H = 1$ if otherwise, D_{critical} is equated using Eq. (8) (Kar & Mohanty, 2006).

$$D_{\mathrm{critical}} = \alpha\sqrt{\frac{N_1+N_2}{N_1N_2}} \quad (8)$$

Assuming that the distribution of the D-values produced is normal and the sizes of $F(\mathrm{x}_1)$ and $R(\mathrm{x}_2)$ are N_1 and N_2 respectively, novelty is indicated when D (Eq. (2)) is above the 2.698σ or the upper quartile of the D distribution. $\alpha = 0.57$ produces the D_{critical} value, for which any values of D beyond or equal to D_{critical} will produce $p < 0.9$.

2.2. Offline and Online Novelty Trending

In order to trend degradation, the capability provided by the previous section must be augmented with the ability to look at parameter distribution change over time. The distributions under comparison should therefore be sampled as two windows of data separated by an appropriate time interval. Two modes of operation are outlined in this paper to provide this measure of change as a function of time:

1. *Offline:* This strategy compares the distributions from the first flight to all other complete flights. In effect, the first flight is used to build a model of normal, and the *offline* test observes the divergence of the system over its lifetime as an analogue to deterioration. Therefore, the analysis performed compares how the subsequent cycles differ from the first cycle: N_1 = number of samples in the x-th cycle and N_2 = number of samples in the 1-st cycle. This methodology will only detect deterioration at a period of complete flights.
2. *Online:* A sliding window approach is employed to enable un-delayed detection and scoring of novelties, therefore allowing indication of novelty occurring during a flight. The sliding window approach is further discussed in the next section.

2.3. Online Trending the Changes to the Distributions: The Sliding Window Approach

The online strategy addresses the trending of novelty by accumulating parameter distribution changes occurring in a time period much less than the typical prognostic horizon of the system degradation. It has been observed that the frequency of distribution changes is indicative of the deterioration for the failure modes explored in this paper. Measures of this trend are termed 'health metrics' and are calculated in two ways: as an average probability of change and as a count of changes per cycle.

The construction of the health metrics involves first applying the multivariate Kolmogorov-Smirnov test to two consecu-

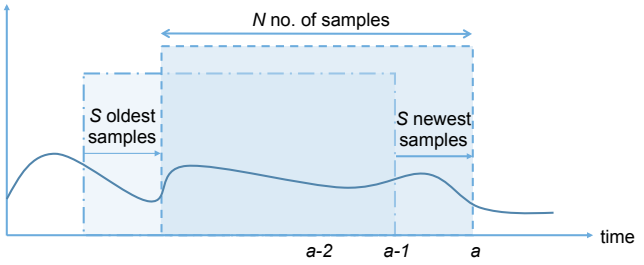


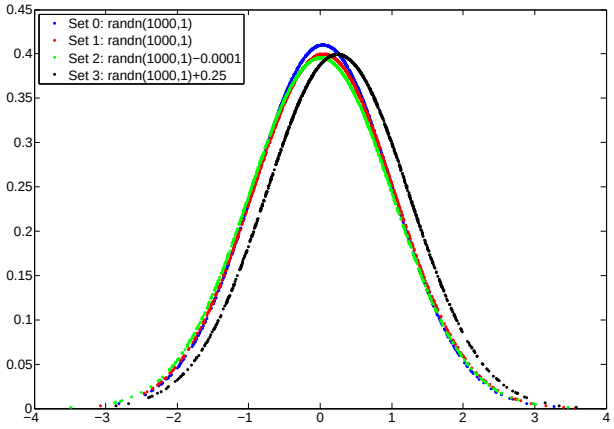
Figure 1. Sliding window. The p values are calculated at a by comparing the distributions of N samples at $a-1$ and at a when the oldest S samples values are replaced with S newest values.

tive sliding windows of data containing N number of samples. The two sliding windows, separated by S number of samples, are used to construct individual multivariate ECDFs (Eq. (6)). The change in distribution is then constructed. The first p probability is calculated when the first and second distributions are obtained, with $N+S$ number of samples, and are subsequently calculated at every interval a when S new samples are obtained. This is as illustrated in Fig. 1.

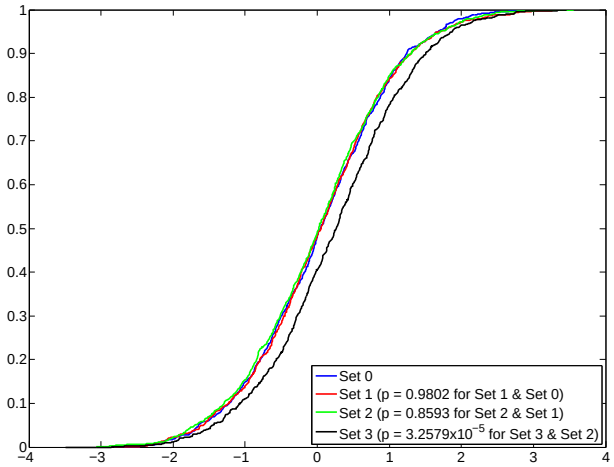
Figure 2 illustrates the concept on synthetic data. The four distributions in the figure are each generated from a window of $N = 1000$ samples selected at different times from Gaussian distributed data with time-increasing mean offset, (Fig. 2a). Formation of an ECDF from the data and calculating the maximum distance D (Eq. (2)) allows changes in distributions to be indicated by the p probability calculated using Eq. (3). A comparison between Set 0 and 1, shows a high similarity ($p = 0.98$), becoming progressively lower as the distance between distributions increases. The lowest p value (approximately 0) occurs when a significant change in the distribution is indicated between Set 2 and 3. Indicating and trending the changes in distribution are useful to identify the deteriorating conditions of the system.

Equations (3)–(4) show the relationship between the D value and its associated p probability of distribution change. The p value decreases exponentially with the increase in the D-value, therefore only when a significant change in the distribution is detected will there be a decrease in the probability of similar distributions. The confidence of novelty is the compliment of the probability given by the p value (i.e. $1-p$). When the system is at nominal conditions and at non-transient operations, the change in distribution should be minimal, with the probability of change given by Eq. (3), $p \geq 0.90$.

The p value, therefore, can be used to visualise the measure of health for the system at any given time. The trend in p may be observed by calculating the running average of the p values at every a during the period of interest (for example a flight), Eq. (9).



(a) The Gaussian PDF of the generated data.



(b) The respective ECDFs and the associated p values when a set is compared against another set.

Figure 2. Kolmogorov-Smirnov p values (Eq. (3)) indicating the change in the distributions.

$$p_{rAve}(a) = \frac{\sum_{i=1}^{a} p(i)}{a} \tag{9}$$

When no novelty is occurring in the system, $p_{rAve} \approx 1$. The value of p_{rAve} decreases with the increase in the rate of novelty detection. Trending the change to the p_{rAve} value shows the severity of the system degradation.

2.4. Nominal and Non-Transient Operations

The nominal and non-transient phase of operations are only investigated at presence. This is to enable the proof of concept of the novelty scoring ability for the proposed method. Furthermore, current aircrafts use an Aircraft Condition Monitoring System (ACMS) to acquire the data for the EHM, and the acquisition of the data is perform at the three defined

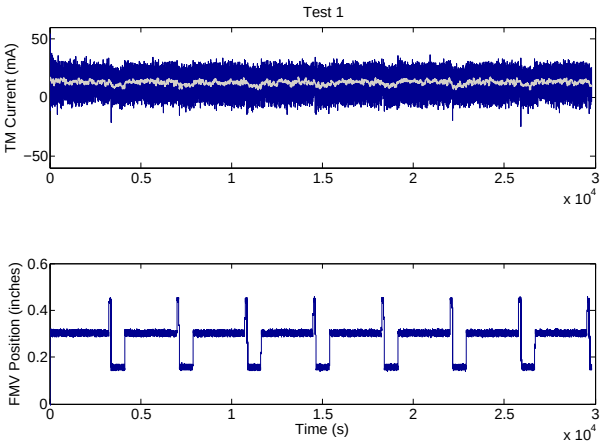


Figure 3. Test 1 (Baseline): Minimal contaminant detected. Small changes in the mean of the TMC are indicated for this test. The mean is indicated by the grey line. The FMV position is averaged at 0.3 inches

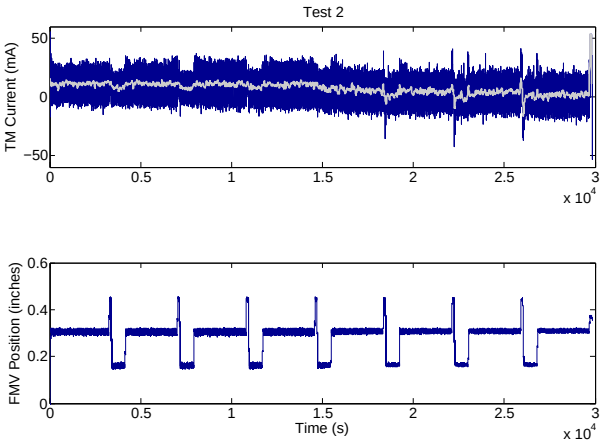


Figure 4. Test 2: Contaminant causing stiction. There are changes to the TMC mean and the FMV positions as the result of the degradation.

phases: take-off, cruise and landing. The EHM then summaries the health of the engine at these phases separately (Waters, 2009). Therefore, we envisage the novelty scoring to be calculated separately at each of these phases. Future work will include understanding the data distribution trends when operating at the transient phase, and to derive the novelty scoring metrics for the nominal and transient operations.

3. EXPERIMENT: NOVELTY DETECTION OF FUEL METERING UNIT

The presented equipment health monitoring scheme is used to detect and trend the degradation of a gas turbine fuel metering

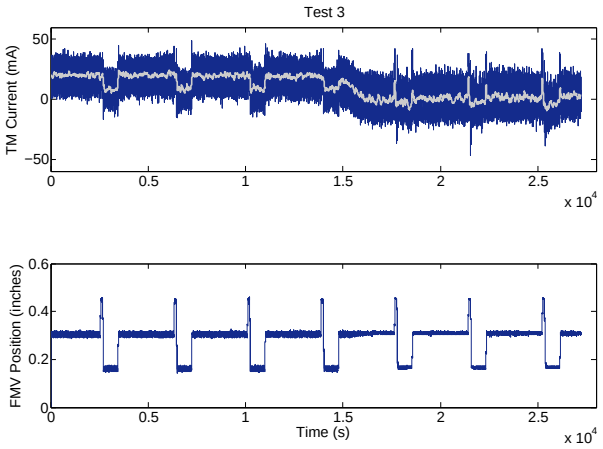


Figure 5. Test 3: Contaminant resulted in stiction. Changes to the TMC mean and FMV positions are shown with the increase in the contaminant level.

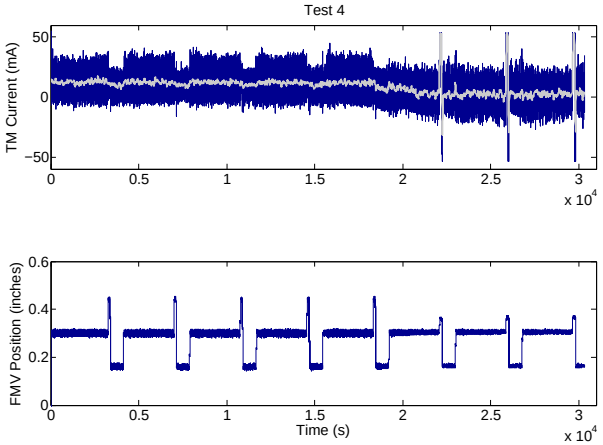


Figure 6. Test 4: Contaminant causing stiction.

unit (FMU). The primary function of a FMU is to regulate fuel flow in response to the Electronic Engine Control (EEC) demand required to deliver commanded engine thrust. This is achieved through position control of two-stage servo fuel metering valve (FMV), which alters the pressure drop across the valve and flow rate through it.

The functional failures associated with the FMU are the loss of FMV bandwidth with poor demand tracking, leading to the inability to control valve position and fuel flow. These, as indicated in Section 1, may be due to debris ingestion resulting in valve friction/stiction or filter clogging.

Data has been collected from fuel system rig tests, which were subject to the introduction of fuel contaminant, and run over up to 8 cycles of cruise, idle and take-off phases. These

Table 1. Mass (as percentage of maximum test amount) of contaminant introduced per test cycle.

Cycle #	Test 1: Baseline	Test 2	Test 3	Test 4
1	13.29	2.85	12.03	0.00
2	18.99	0.95	4.43	4.11
3	22.15	5.06	3.80	2.85
4	20.25	8.54	7.59	7.59
5	24.05	9.49	11.71	4.75
6	36.39	37.34	23.10	19.30
7	35.44	43.04	28.80	34.18
8	69.62	100.00	31.96	98.73

tests exhibited functional failures from loss of metering valve control at high contaminant levels and serve as a basis for evaluating the outlined novelty trending schemes.

The EHM scheme presented is used to indicate how the system degrades as the result of the contaminant introduction. At present, the analysis of the scheme is to indicate the degradation only when the engine is supposedly at the cruise phase. This is shown in Figs. 3–6 when the FMV position is averaged at 0.3 inches. In all three tests, the mean at cruise of the TMC reduces as the control system compensates for the effects of the increase in the contaminant level.

The mass of contaminant introduced in each cycle is listed in Table 1. It should be noted this is not a measure of degradation, and only indicates the mass of particles introduced to the system at each cycle presented as a percentage of maximum cycle dosage over all tests.

Two signals are initially chosen for use to monitor the degradation level in response to the introduction of the contaminant. They are:

1. The torque motor current (TMC), and
2. The fuel metering valve (FMV) position.

The TMC values and the FMV position values are sampled at 40Hz, and are normalised (Eq. (10)) so that their values are between -1 to +1 prior to the analysis.

$$x_n = (b - a) \times \frac{x_o - x_{min}}{x_{max} - x_{min}} + a \qquad (10)$$

x_n is the normalized value and x_o is the value to be normalized. a and b are the minimum and maximum value of the range to be normalized to, which in this case is $a = -1$ and $b = +1$. x_{max} and x_{min} are the maximum and minimum values of the range of x_o.

Figure 3 indicates the values of these variables when the majority of the contaminants introduced are captured by the low pressure (LP) filter. The LP filter traps the contaminant upstream of the metering valve, therefore preventing stiction and degradation. Physical analysis of this test also indicates that only a small amount of contaminant is detected in the FMU as the results of the filtering, too small to cause degradation. Because of this, minimal changes in the system dynamic response are shown, despite the contaminant introduction. This test acts as the baseline test (Test 1: Baseline) to evaluate the capabilities of the presented EHM scheme.

In tests 2–4 (Fig. 4–6), the system degrades over time and with the increase in the contaminant level introduced per flight cycle.

For the sliding window approach, three different window sizes are considered: 60 seconds of data, $N = 2400$ number of samples; 120 seconds of data, $N = 4800$ samples; and 300 seconds with $N = 12000$ samples. The distribution of the sensors values are updated and compared when the oldest S sample values were replaced with the newest S values, at every a. Six different sets of N and S are analysed:

1. For 60 and 120 seconds of data ($N = 2400$ and $N = 4800$ samples): $S = 40$ samples (1 second of data).
2. For 60 and 120 seconds of data ($N = 2400$ and $N = 4800$ samples): $S = 80$ samples (2 seconds of data).
3. For 300 seconds of data ($N = 12000$ samples): $S = 200$ samples and $S = 400$ samples (5 seconds and 10 seconds, respectively).

4. RESULTS

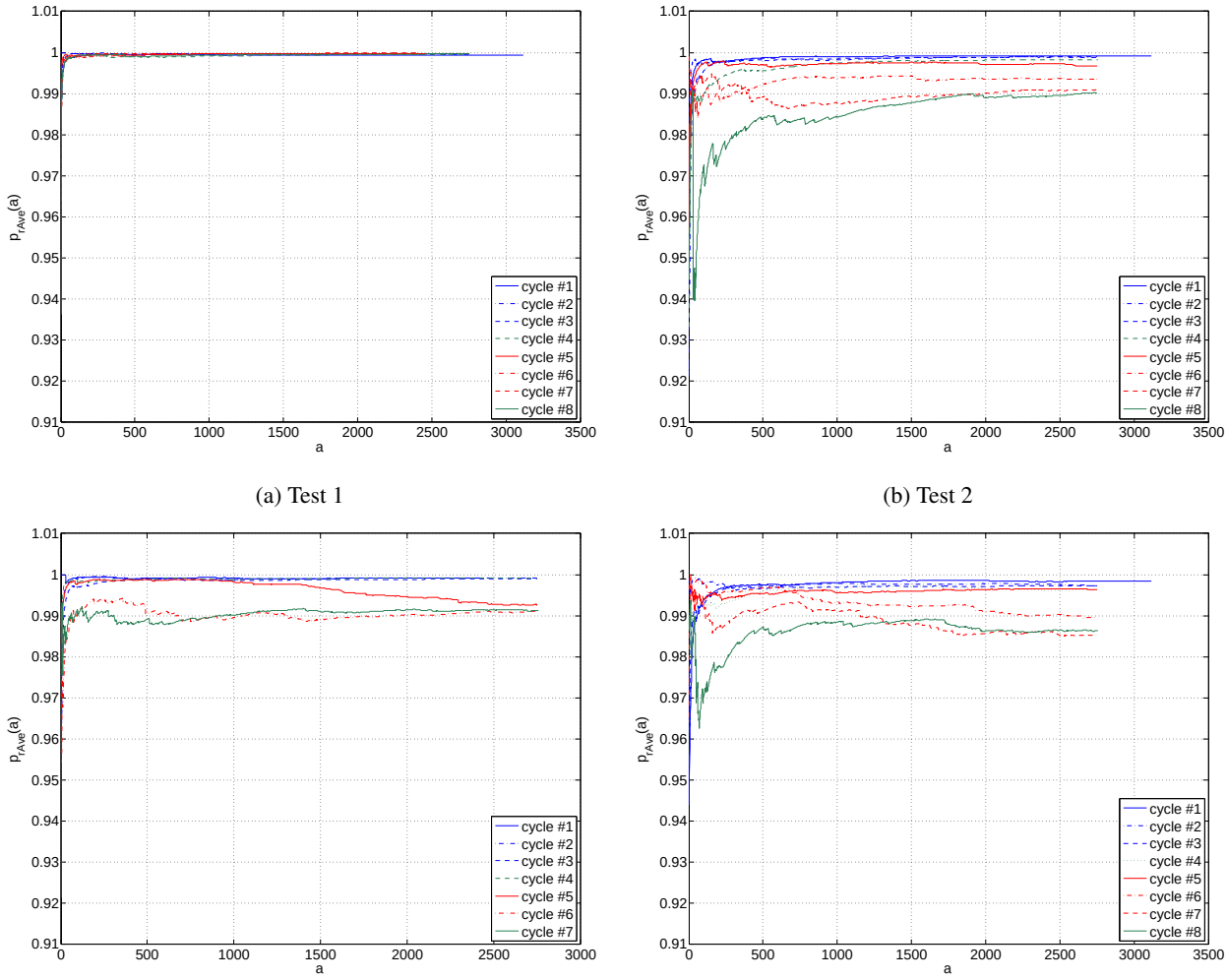
4.1. Offline novelty trending

Figure 7 represents the D values produced by the analysis using the offline strategy comparing each subsequent cycle to the first cycle. The figure shows, for all tests other than the baseline (i.e. not Test 1), a large change to the cycles' distributions are shown when they are compared to their initial cycle. The significance in change is indicated by the large increase in the D value for each cycle, caused by the change to the FMU system dynamics. The D-values remain approximately the same for Test 1: Baseline.

The D values are directly used in the offline analysis. The significant changes between each cycle result in large D values being produced by the comparative analysis, these all result in the the probability of no change tending to a very small value ($p \to 0$). Therefore, for the offline cycle-to-cycle mode of comparison, the novelty detection is made based on the D values instead of its $p-$values.

4.2. Online novelty trending

This section evaluates the performance of the two on-line trending approaches as described in Section 2.3. Tables 2–4 show the number of changes to the distributions cycle (i.e. the count of evaluations when $p < 0.90$). Low detection rates are shown for Test 1: Baseline, as most of the contaminants are filtered prior to the metering valve. Frequent changes in distribution is shown by higher count values when the con-



(a) Test 1

(b) Test 2

(c) Test 3

(d) Test 4

Figure 8. The EHM health metric provided by the $p_{rAve}(a)$ values with $N = 2400$ samples (60 seconds of data) and $S = 40$ samples (1 second of data).

Table 2. Number of occurrences when $p < 0.90$, when $N = 2400$ number of samples (60 seconds of data).

Cycle #	N = 2400 samples, S = 40 samples				N = 2400 samples, S = 80 samples			
	Test 1: Baseline	**Test 2**	**Test 3**	**Test 4**	**Test 1: Baseline**	**Test 2**	**Test 3**	**Test 4**
1	**0**	**0**	**0**	**0**	144	168	110	294
2	**0**	**0**	**0**	**0**	100	177	117	332
3	**0**	**0**	**0**	**1**	59	190	148	356
4	**0**	**0**	**0**	**2**	100	216	110	352
5	**0**	**17**	**59**	**1**	57	223	354	379
6	**0**	**53**	**70**	**81**	79	314	384	395
7	**0**	**72**	**68**	**128**	79	352	454	452
8	**0**	**84**	**N/A**	**114**	81	290	N/A	376

taminants were not filtered from the unit (Tests 2–4).

The two tables show that the optimal N:S ratio for the EHM scheme is 60:1 (indicated in bold). Any increase to the ratio will result in a higher number of false detection, i.e. higher number of false detection for the baseline test (Test 1) when alternative ratios are used. Results also show that the window with 60 seconds of samples ($N = 2400$ samples) and 1 second interval ($S = 40$ samples) is sufficient for detection of

Table 3. Number of occurrences when $p < 0.90$, when $N = 4800$ samples (120 seconds of data).

Cycle #	N = 4800 samples, S = 40 samples				N = 4800 samples, S = 80 samples			
	Test 1: Baseline	**Test 2**	**Test 3**	**Test 4**	**Test 1: Baseline**	**Test 2**	**Test 3**	**Test 4**
1	0	0	0	0	**20**	**20**	**13**	**57**
2	0	0	0	0	**19**	**24**	**15**	**73**
3	0	0	1	0	**8**	**30**	**19**	**72**
4	0	0	1	0	**9**	**34**	**15**	**87**
5	0	0	35	1	**3**	**50**	**120**	**84**
6	0	0	40	51	**10**	**87**	**122**	**128**
7	0	2	38	58	**8**	**95**	**180**	**122**
8	0	6	N/A	55	**10**	**94**	**N/A**	**116**

Table 4. Number of occurrences when $p < 0.90$, when $N = 12000$ samples (300 seconds of data).

Cycle #	N = 12000 samples, S = 200 samples				N = 12000 samples, S = 400 samples			
	Test 1: Baseline	**Test 2**	**Test 3**	**Test 4**	**Test 1: Baseline**	**Test 2**	**Test 3**	**Test 4**
1	**25**	**48**	**36**	**82**	61	76	58	106
2	**15**	**46**	**15**	**92**	37	75	39	105
3	**15**	**48**	**32**	**89**	37	73	60	99
4	**11**	**46**	**12**	**81**	37	77	33	89
5	**11**	**43**	**114**	**99**	33	64	128	114
6	**16**	**62**	**105**	**92**	32	86	124	107
7	**7**	**78**	**160**	**123**	28	92	155	116
8	**10**	**74**	**N/A**	**107**	35	98	N/A	111

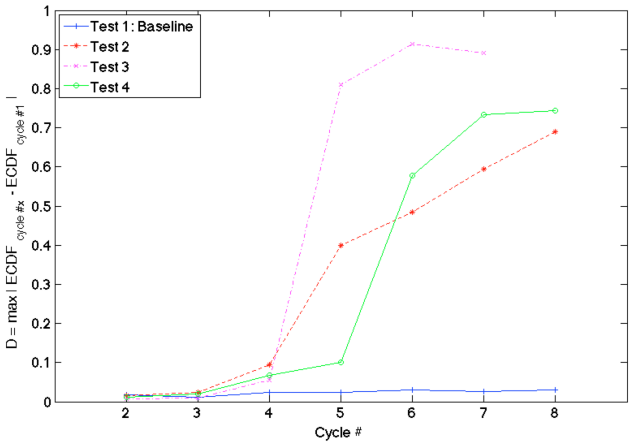


Figure 7. The D values when comparing a cycle to its first cycle.

the degradation. This because of the no (zero count) detections for Test 1: Baseline, as well as the ability to detect the changes to the TMC and FMV positions for Test 2–4.

The alternative health metric, $p_{rAve}(a)$, for each test for $N = 2400$ samples and $S = 40$ samples is shown in Fig. 8. For all non-baseline test (Test 2–4), the values of the $p_{rAve}(a)$, which is an indicative of the health of the system, reduces overtime. This shows that the health of the system has degraded with time with the increase in contaminant per flight cycle. $p_{rAve}(a)$ are constant and are ≈ 1 for Test 1: Baseline, indicating no system degradation because the LP filter has trapped the contaminant upstream of the metering valve, therefore preventing degradation. The decrease in health also indicates the increase in the novelty detection rate.

4.3. Univariate vs Multivariate

As indicated in Section 1, a gas turbine is complex electro-mechanical system. Determining the most effective parameter for analysis is not always apparent. If one is to perform univariate analysis, the incorrect selection of sensing parameter will lead to a different outcome. For example, if one chooses the FMV position to indicate novelty for N = 2400 samples (60 seconds worth of data) and S = 40 samples (1 second worth of data), as shown in Table 5, no trending of degradation is achievable. Similar observation is shown when the analysis is performed using the TMC's distributions for N = 2400 and S = 40. The short time interval between time windows is not sufficient to identify the changes in these co-related variables when they are treated in isolation.

The analysis presented earlier in this paper is for bivariate analysis ($V = 2$ in Eq. (6)). Figure 9 and Table 6 show the results when increasing the number of variables, V, analysed from the measured rig test data, at the optimal N:S ratio of 60:1 (N = 2400 samples of data and S = 40 samples). The last recorded p_{rAve} values for $V = \{3, 4, 5\}$, i.e. the cycle average p value, decrease with the increase in the level of contaminants for all non-baseline tests (Fig. 9). The detection event count also increases with the increase in contaminants (Ta-

Table 5. Number of occurrences when $p < 0.90$, when $N = 2400$ samples and $S = 40$ samples.

Cycle #	Variable # 1 TMC				Variable # 2 FMV position			
	Test 1: Baseline	**Test 2**	**Test 3**	**Test 4**	**Test 1: Baseline**	**Test 2**	**Test 3**	**Test 4**
1	0	0	0	0	33	11	15	8
2	0	0	0	0	23	11	41	7
3	0	0	0	0	18	5	33	15
4	0	0	0	0	23	19	28	26
5	0	0	4	0	14	88	49	21
6	0	0	4	4	22	97	14	153
7	0	2	0	4	7	52	39	14
8	0	0	N/A	4	22	7	N/A	12

Table 6. Number of occurrences when $p < 0.90$ for the univariate test with $N = 2400$ samples and $S = 40$ samples.

Cycle #	$V = 3$ (+ MV Downstream Pressure)				$V = 4$ (+ LP Pump Outlet)				$V = 5$ (+ SPR Pressure)			
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 1**	**Test 2**	**Test 3**	**Test 4**
1	3	0	0	0	8	3	0	0	6	2	0	0
2	1	1	0	1	0	1	0	0	0	1	0	0
3	0	2	0	0	0	0	0	1	0	0	0	1
4	0	5	0	3	0	4	2	2	0	4	1	1
5	0	73	72	2	1	17	16	0	2	75	25	0
6	0	159	107	118	0	48	19	27	0	119	32	59
7	1	307	99	163	0	120	18	57	1	290	39	118
8	3	366	N/A	158	1	226	N/A	56	2	396	N/A	118

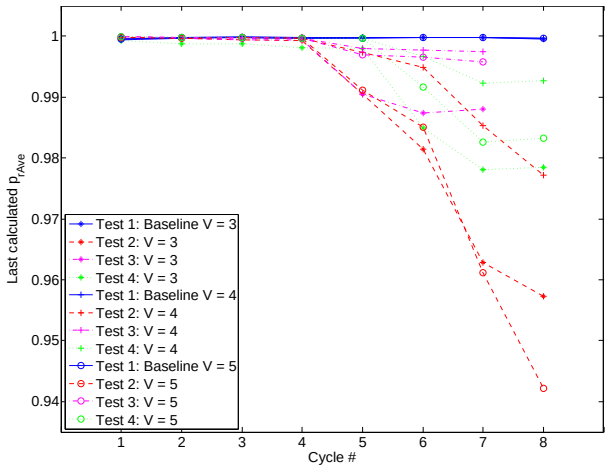


Figure 9. The last $p_{rAve}(a)$ calculated at the end of each cycle for the multivariate analysis related to Table 6.

ble 6). Additional variables included are, for $V = 3$, the normalised metering valve (MV) downstream pressure sensor values, and, for $V = 4$, the fourth variable is the normalised low pressure (LP) supply pressure data. The third analysis is performed with $V = 5$, which adds the servo-pressure.

An analysis of the sensitivity to degradation from these results can be made with respect to the physical interpretation of Figure 9. The average p value for tests where $V = 3$ or $V = 5$ are consistently lower (a change, thus degradation, more likely) than for the test with 4 variables. From this, we conclude that adding the LP supply pressure parameter (in $V = 4$) makes the change to multi-variate distributions less significant, not adding to the ability to determine degradation. The LP pump, thought to be robust to containment itself, is upstream from the valves which are impacted by the containment and therefore is not affected by system degradation. On the other-hand, servo-pressure is controlled by an additional valve to the FMV and therefore introduces sensitivity to another element of the system. The downstream flow pressure (added in $V = 5$) is dependent on the supply pressure (affected by a spill valve) and the valve position, again this valve introduces another candidate source of degradation from the spill valve. It is plausible that these observations could be used to aid fault isolation in future work.

These results corroborate the hypothesis that a combination of multiple sensing parameters is powerful for novelty detection analysis and health scoring of a system, as more dynamics are captured as part of the analysis. The non-parametric two-sample Komolgorov-Smirnoff test provides the mechanism to perform multivariate analysis with minimal pre- or post-processing of the provided data (aside from the normalisation of the data so that their values are between -1 to 1).

5. CONCLUSION

This paper presents the results of a multivariate equipment health monitoring (EHM) scheme that utilises non-parametric statistics. The scheme was developed to provide early detec-

tion of the gas turbine fuel metering valve faults, and to enable the scoring of the significance of the degradation. Degradation assessment can inform maintenance planning and consequently reduce disruption.

The scoring is achieved using non-parametric statistics that approximates the data distribution for the time window consisting of the current and the previous samples. The data distribution estimate adapts over time, and a generic measure of difference, a multivariate two-sample Kolmogorov-Smirnov test, is shown to provide diagnosis capabilities. The equipment health monitoring scheme is able to trend the degradation of the fuel metering valves, degradation resulted from the varying levels of contaminant introduced to the engine. Results indicate that the level and rate of detection increases with the increase in the contaminant level, which resulted in the degradations.

As indicated in Section 2.4, the analysis is restricted to cruise phase or at the non-transient phase of flight operations. This is to enable us to present the proof-of-concept capabilities of the novelty scoring metric using the non-parametric multivariate two-sample Komolgorov-Smirnoff test. Future work will include, but not limited to, the analysis of the capabilities of the algorithm to cope with transient phases. We envisioned that a different scoring metric is required to indicate for novelty when the system is in the transient phase of operations.

Two methods for novelty trending are presented in this paper: online sliding window approach and the offline cycle-by-cycle approach (Section 2.2). Schemes to fuse the outputs of these approaches together, along with schemes to trend the outputs over time, may provide advantages in detecting different failure modes and will be investigated.

As presented, the use of multivariate two-sample Kolmogorov-Smirnov test for the EHM scheme simplifies and enhances novelty detection, eliminating the need to choose variables or summary statistics for health analysis prior to system deployment.

ACKNOWLEDGMENT

The authors would like to thank Rolls-Royce plc for their support and direction of this research.

REFERENCES

Andrade, F. A., Esat, I., & Badi, M. N. M. (2001). A new approach to time-domain vibration condition monitoring: Gear tooth fatigue crack detection and identification by the kolmogorov-smirnov test. *Journal of Sound and Vibration*, *240*(5), 909-919. doi: 10.1006/jsvi.2000.329

Eleffendi, M. A., Purshouse, R., & Mills, A. R. (2012). Gas Turbine Fuel Valve Diagnostics. In *Proceedings of 2012 ieee aerospace conference.*

Greenwell, R. N., & Finch, S. J. (2004). Randomized rejection procedure for the two-sample kolmogorov-smirnov statistic. *Computational Statistics & Data Analysis*, *46*(2), 257-267.

Hall, L. D., & Mba, D. (2004). Acoustic emissions diagnosis of rotor-stator rubs using the ks statistic. *Mechanical Systems and Signal Processing*, *18*, 849-868. doi: 10.1016/S0888-3270(03)00050-5

Kar, C., & Mohanty, A. R. (2006). Multistage gearbox condition monitoring using motor current signature analysis and kolmogorov-smirnov test. *Journal of Sound and Vibration*, *290*(1-2), 337-368. doi: 10.1016/j.jsv.2005.04.020

Marsland, S. (2003). Novelty detection in learning systems. *Neural Computing Survey*, *3*, 157-195.

Modenesi, A. P., & Braga, A. P. (2009). Analysis of time series novelty detection strategies for synthetic and real data. *Neural Processing Letters*, *30*(1), 1-17. doi: 10.1007/s11063-009-9106-4

Salgado, D. R., & Alonso, F. J. (2006, February). Tool wear detection in turning operations using singular spectrum analysis. *Journal of Materials Processing Technology*, *171*, 451-458.

Scheffer, C., & Heyns, P. S. (2001). Wear monitoring in turning operations using vibration and strain measurements. *Mechanical Systems and Signal Processing*, *15*(6), 1185-1202.

Sohn, H., Farrar, C. R., Hunter, N. F., & Worden, K. (2001). Structural health monitoring using statistical pattern recognition techniques. *Journal of Dynamic Systems, Measurement, and Control*, *123*(4), 706-711. doi: 10.1115/1.1410933

Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., & Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd international conference on very large data bases (vldb '06)* (p. 187-198).

Waters, N. (2009, June). Engine health management. *The Ingenia Magazine*(39), 37-42.

Zhan, Y., & Mechefske, C. K. (2007). Robust detection of gearbox deterioration using compromised autoregressive modeling and kolmogorov-smirnov test statistic-part i: Compromised autoregressive modeling with the aid of hypothesis tests and simulation analysis. *Mechanical Systems and Signal Processing*, *21*, 1953-1982.

BIOGRAPHIES

Maizura Mokhtar holds a BEng (Hons) in Electronics and Computing from Nottingham Trent University, an MSc(Eng) in Control Systems from the University of Sheffield and a PhD in Electronics from the University of York. She has previously worked as a Research Associate in the Department of Electronics at the University of York, and a Postdoctoral Research Assistant in the School of Computing, Engineering and Physical Sciences, University of Central Lancashire (UCLan). Her research interests focus on the development of intelligent autonomous systems, notably the use of computational intelligence algorithms (artificial neural networks, evolutionary algorithms, artificial immune systems) for achieving intelligent and adaptive behaviours in engineering systems.

Joseph C Edge, CEng is a Fuel & Heat Management Specialist at Rolls Royce. He graduated from Warwick University with a BEng System Engineering in 1995. He worked in the automotive industry for 8 years as an Aerodynamicists and Cooling Airflow Engineer before his current role at Rolls Royce.

Andrew R Mills, CEng is a Research Fellow at the University of Sheffield, from where he graduated with a MEng in Control System Engineering. He has worked in the defence industry on aerospace and automotive applications before his current role as Programme Manager and leading Health Management Technologies research within the University Technology Centre supported by Rolls-Royce. Research interests are in a broad range of aspects related to gas turbine system health management and controls.

A Generalized Machine Fault Detection Method Using Unified Change Detection

Wenyi Wang, David Forrester and Peter Frith

Aerospace Division, Defence Science and Technology Organisation, Melbourne, VIC 3207, Australia

Wenyi.Wang@dsto.defence.gov.au
David.Forrester@dsto.defence.gov.au
Peter.Frith@dsto.defence.gov.au

ABSTRACT

Many different techniques have been developed for detecting faults in rotating machinery. This is because different fault types typically require different techniques for the effective detection of the fault. However, for many new or unknown fault types, we have found that the existing detection techniques are either incapable or ineffective, and that we therefore need to come up with brand new methods after the fault event. This can significantly constrain the usefulness and effectiveness of Prognostic Health Management (PHM) systems. In this paper we attempt to look at detecting global changes in the synchronously averaged signals as the machine's health status progresses from healthy to faulty, and to define one unified signal processing technique and its associated condition indicators for the detection of changes caused by various types of faults in rotating machinery. The proposed method is conceptually very simple, and its effectiveness is demonstrated using vibration data from machines with several different types of faults. The results have shown that this single unified change detection approach can be very effective in detecting and trending changes caused by many different types of machine faults.

1. INTRODUCTION

Since the advent of some benchmark technologies, namely the envelope technique for bearing diagnosis in early 1970's by Burchill *et al* (1973) and the time synchronous averaging technique for gear diagnosis in mid to late 1970's by Braun (1975) and Stewart (1977), the field of machine diagnostics has had enormous advancement. Over the last four decades, many techniques have been developed for detecting various types of faults in rotating machinery (e.g. Forrester 1996, McFadden 2000, Wang 2001 and techniques discussed in the review papers by Randall 2011 and Lei 2013, etc.). However, it is typically found that different techniques are required for the effective detection of new types of fault. This need to specifically develop new methods whenever a new type of fault arises can significantly constrain the usefulness and effectiveness of PHM systems, especially for new platforms such as the JSF where the PHM capability is designed in during the early stages of development.

In general, for gear tooth related local faults we tend to employ the residual signal after removing the gear mesh harmonics in the spectrum of synchronous signal averages (Stewart 1977, Forrester 1996, Wang 2001). For a localized bearing fault we will most likely look at the resonance demodulation technique (Burchill 1973, Wang and Harrap 1996). For other common faults like rotor unbalance and shaft misalignment we may try to find changes in the low shaft orders such as the first three orders (Forrester 1996, Larder 1999, Vecer *et al* 2005). In cases of spline or pump faults, we will probably focus on the changes at the relatively higher shaft orders or the pump characteristic frequency and its harmonics (Galati 2007, Becker 2007, Hancock 2006). For turbine engine disk cracks, the state-of-the-art technology is to use tip timing data analysis to detect this type of fault (Wang and Muschlitz 2010). There are many other fault types that involve specific detection techniques.

These techniques are widely employed in health and usage monitoring systems (HUMS) for helicopters. Unfortunately, when new or unknown types of faults occur these methods are often either incapable or ineffective to detect the faults. In 2002, planet carrier plate cracking was a new type of fault found in the main rotor transmission of the Blackhawk and Seahawk fleets around the world. Several techniques including those by Blunt and Keller (2006) and by Wang

Wenyi Wang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and Keller (2007) were specifically developed for the fault type after the fault events. Note that this carrier plate has been re-designed by the manufacturer and the plates with the new design are being retrofitted into the fleet. A more recent example is the crash of a Super Puma utility helicopter in the North Sea in 2009 caused by a gear/bearing fault that was not detected by the onboard HUMS (Jarvis and Sleight 2011). The investigation indicated that the most likely root cause was a spall-induced fatigue crack in the 2nd stage planet gear/bearing in the main transmission gearbox of the helicopter, which propagated though the gear/bearing body and led to the disintegration of the gear/bearing, causing the catastrophic failure of the gearbox. The fatal accident led to the loss of sixteen lives.

There have been previous attempts to develop more versatile methods for detecting various types of faults. These include the parametric model-based approach by Wang and Wong (2000) and Wang (2008) in building a linear prediction model for the healthy-state signal and then using this model as an inverse filter to process the future-state signals. The method was proven effective in many cases, especially in the case of multiple gears on the same shaft, but a consistency problem with the selection of model orders can show up when peculiar perturbations exist in the signal. This is probably due to the nature of parametric modeling and lack of constraints in the optimization process. In other words the method lacks robustness. Other studies were carried out by Man *et al* (2012) to use a versatile sinusoidal model for fault diagnosis in a more robust manner, and by Galati *et al* (2008) to use a generalized likelihood ratio algorithm for detecting bearing faults in helicopter transmissions. The work carried out by Lee (2010) was an attempt of detecting a general class of faults using correlation algorithms in a low cost HUMS.

In this paper we attempt to look at detecting global changes in the vibration signals as a machine's health status progresses from healthy to faulty for various different types of faults, and to find one unified signal processing technique and its associated condition indicators for the detection of these changes. The detection of changes due to machine faults often involves comparison of signals from the healthy-state to the faulty-state of the machine. However, a direct comparison in the time domain is often prohibited simply because these signals are in most cases not phase-aligned. Our unified approach deals with the synchronously averaged or re-sampled vibration signals from a rotating component in the machine as it progresses from a healthy state to a faulty state. The healthy-state signal x is employed as a reference, and it is phase shifted by the phase difference from the future-state (healthy- or faulty-state) signals y. The shifted healthy-state signal x_s is then subtracted from future-state signals y to form the change signals. We expect that fault-induced changes will be captured by the change signal. Statistical measures can then be derived from the change signal as condition indicators, and trended over time for fault detection purposes.

The technique is conceptually very simple, and its effectiveness is demonstrated in the paper. Vibration data from machines with several different types of faults are used for the demonstration. The fault types include gear tooth cracks in simple gearboxes; non-uniform gear tooth wear and vane pump failure in turbo-machinery; and nut looseness and planet carrier plate cracking in helicopter transmission systems. The results show that this single unified change detection approach can be very effective in detecting changes caused by many different types of machine faults. We anticipate that further adaptation and validation of this approach may lead us towards a universal method for fault detection in rotating machinery, including faults in gears, bearings, rotors and pumps.

The main driver of developing such a unified approach is to equip existing and future HUMS and PHM systems with the capability of detecting new and unknown types of faults. The implementation of the proposed technique into an existing health monitoring system should be straight-forward.

2. BACKGROUND OF SIGNAL ALIGNMENT

In gear fault diagnosis, we may tend to assume that the synchronously averaged signals are phase aligned if a tachometer signal is employed as a phase reference signal for the rotating components in the gearbox. However, in many cases, the use of a pulsed phase reference signal means that the zero crossing point (phase alignment point) can only be determined to within one sample point, i.e. the rising edge of the pulse occurs somewhere between two sample points. This means the signal averages are only aligned to within one sample point at the original sampling frequency. Note that if the speed reference were a sinusoidal waveform, the zero crossing point can be determined to greater accuracy by the use of interpolation. Additionally, there may be other error sources in the phase reference signal, such as the speed-dependent pulse amplitude, which may cause the misalignment of averaged signals by more than one sample point.

Taking gear tooth cracking as an example fault type, we will start with two actual signals acquired in a gear tooth crack propagation test conducted at the Defence Science and Technology Organisation (DSTO), Australia (Forrester 1996, Vavlitis 1998). This test series will be described in more detail in Section 5.1. Then we will look at some simulated gear mesh signals to see the necessity for accurate signal alignment and some of the problems that can occur when conducting this alignment.

2.1. Example of Gear Mesh Signal Alignment by Direct Signal Shifting

Figure 1 shows two signals of gear mesh vibration, which came from a spur gear at two stages of tooth cracking. The first one is labeled G6b.071 (signal *x*) where the crack was probably just initiated from the stress-riser notch, i.e. there was no visual indication of crack but the post-test fractography analysis showed an equivalent through-crack length of about 0.7mm. The other signal is G6b.110 (signal *y*) where the tooth crack length was around 50 percent of the tooth width (2.75mm by visual inspection from side, about 3.15mm by fractography analysis). Note that the total length of the projected crack path was 5.82mm for this gear. These two signals have very similar amplitude and their phases are not perfectly aligned. We estimated the phase difference by using maximum cross correlation coefficient to the accuracy of one sampling period, and found that the phase difference corresponds to about 3 sample points. This near-integer-sample phase shift is likely to be due to the on-line angular data acquisition of the G6 test data triggered by the TTL pulses (0-5 volts square pulse, 1024 pulses/rev) of an optical shaft encoder, where each averaged signal might have started from a slightly different TTL pulse. However, this phase shift (i.e. the number of samples) may be different from signal to signal.

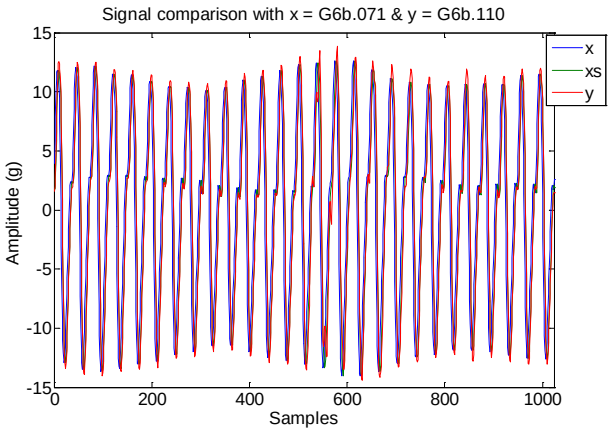


Figure 1. Comparison of synchronous signal averages of gear mesh vibration

The signal *x* is then shifted by 3 samples and the shifted version is denoted by x_s. As we can see in Figure 2, x_s is well aligned with signal *y*. A straight subtraction of x_s from *y* then produces the so-called change signal δ, as shown in Figure 3. Obviously, the change signal has picked up the changes caused by the tooth cracking. It has a kurtosis, as defined in Eq. (12) of this paper, value of 9.3 where a kurtosis value of 3.5 would typically be regarded as an indication of an early localized fault. This is comparable to some of the benchmark indicators, such as a kurtosis of 5.2 for the residual signal, derived by removing the gear mesh harmonics in the spectrum of the Synchronous Signal Average (SSA), or a kurtosis of 11.4 by further removing the 1st and 2nd sidebands of the harmonics. The residual signal kurtosis is one of most commonly used Condition Indicators (CIs) in gear fault diagnosis.

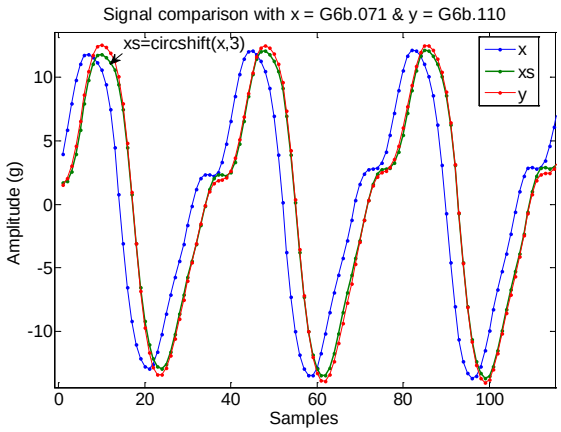


Figure 2. Zoomed version of Fig. 1

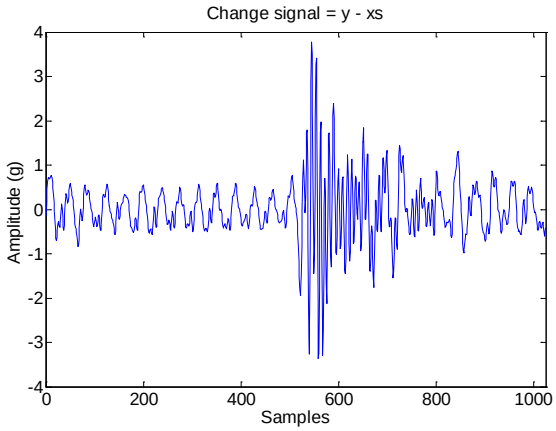


Figure 3. The change signal (full bandwidth): $\delta = y - x_s$, with a kurtosis of 9.3.

Normally, a localized fault tends to cause more changes in relatively high frequency range whereas a distributed fault is more likely to produce changes in low frequency. Therefore, if we view the change signal in two frequency bands, i.e. a low-order band and a high-order band, where the cross-over occurs at 85 shaft orders (which is just above the 3rd gear mesh harmonic at 3×27 = 81 orders and is below a structural resonance), as shown in Figure 4, we can see that the crack-induced change in the high-order band is far more pronounced than that in the low-order band. The kurtosis values for these two bands are 16.5 and 3.8 respectively.

Intuitively, we can say that the key step here is to align signals acquired in a healthy-state (or reference signal) and a faulty-state (or monitored signal). We can also use the instantaneous phase cross correlation to obtain sharper maxima so that the signal shift amount may be defined more clearly. However, to achieve signal alignment with an

accuracy better than one sample period, we will need to interpolate the cross correlation function.

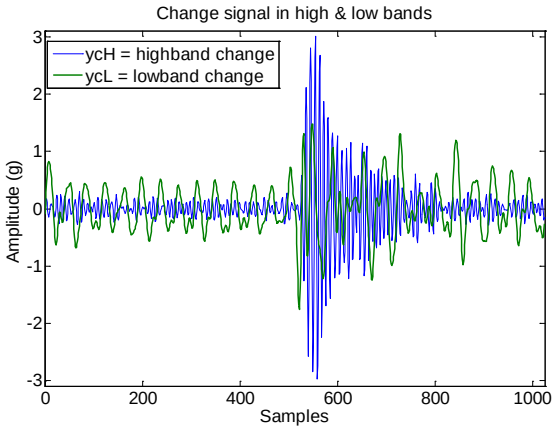


Figure 4. The change signal divided into high- and low-order bands, with the kurtosis values of 16.5 and 3.8 respectively.

2.2. Understanding Signal Alignment using Simulated Signals

We can see, from the above example, that phase alignment of the two signals is essential. Even in cases where signal averaging has been performed using a phase reference signal from the shaft of interest, small variations in signal alignment from one run to the next may occur. This can be caused by a change in the shaft reference probe (e.g. dirt, physical movement of the sensor, etc.), and, more likely, the inherent errors in the tachometer signal processing (e.g. errors from interpolating the position of the zero-crossing between sample points). Also, in cases where the phase reference signal sensor is not physically attached to the shaft of interest, or where the synchronous averaging is carried out using a phase reference directly derived from the vibration signal (Bonnardot *et al* 2005), it is not feasible to phase-align the averaged signals during the synchronous averaging process.

For the remaining part of the paper, we denote a uniform phase shift by $\Delta\theta$ and a uniform time delay by Δt as shown in the following expression. The word 'uniform' applies to multi-frequency signatures, where the phase shift and time delay are the same for all the frequency components.

$$y(t) = \sum_k A_k \cdot \sin\left[2\pi\ f_k(t-\Delta t) + (\theta_k + \Delta\theta)\right]$$

2.2.1. Uniform phase shift

First of all, when talking about phase alignment we may tend to think of aligning the initial phase of the signal. If we have a test signal of

$$x(t) = \sin(2\pi \cdot 27 \cdot t + 0.987)$$

with a sampling rate of 1024 samples/second, i.e. $t = (0{:}1023)/1024$, and an arbitrary initial phase of $\theta = 0.987$ radians, and we then define a phase-shifted version of this signal, $y(t)$, where the initial phase of this frequency component is changed by $\Delta\theta = -0.4975$ radians from $x(t)$, then this phase shift will correspond to almost exactly 3 sample points, i.e. $1024\times0.4975/(2\pi\times27)=3.003$. Therefore, to align $x(t)$ and $y(t)$ we could simply shift signal $x(t)$ by 3 sample points, e.g. using the Matlab function '*circshift*': x_s = circshift(x,3). However, if the phase shift does not correspond to a near-integer sample point, then this alignment process will not work. Figure 5 shows the signal $y(t)$ with a phase shift of –0.57 radians (or 3.4406 samples) and the signal $x(t)$ shifted by 3 sample points. As can be seen, rounding to the nearest sample point does not produce a good result, and a finer (fractional-point) shift resolution is required.

Now let us employ a two-component sinusoid like

$$x(t) = \sin(2\pi \cdot 27 \cdot t + 0.987) + \sin(2\pi \cdot 2 \cdot 27 \cdot t + 1.053)$$

where the 2nd component is a harmonic of the 1st one. Signal $y(t)$ is then defined as signal $x(t)$ shifted by –0.4975 radians at both frequency components (i.e. uniform phase shift). If we now shift $x(t)$ by 3 sample points using '*circshift*' we cannot get a good alignment as shown in Figure 6. This is because the phase shift of –0.4975 radians for the higher frequency component corresponds to almost 1.5 sample points instead of 3 for the lower frequency component, i.e. $1024\times0.4975/(2\pi\times2\times27) = 1.5015$.

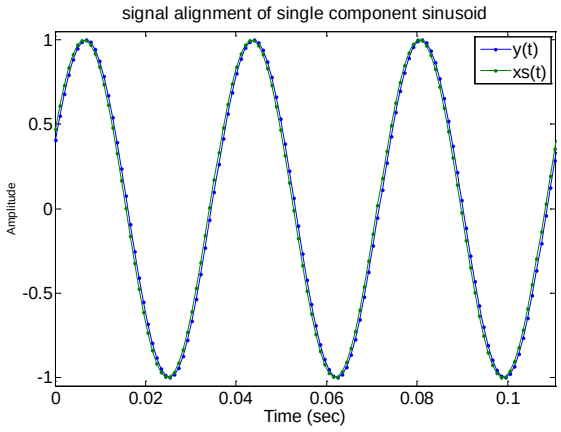


Figure 5. Simulated signal x_s & y showing that direct signal shifting would not work with a phase shift of non-integer sample.

There are two important observations from this section: (1) direct signal shifting by integer sample points would not be a good approach if the phase difference does not give a time delay corresponding to integer number of data samples; (2) direct signal shifting is also no good for multiple components signals where the phase shift is the same across all the frequency components.

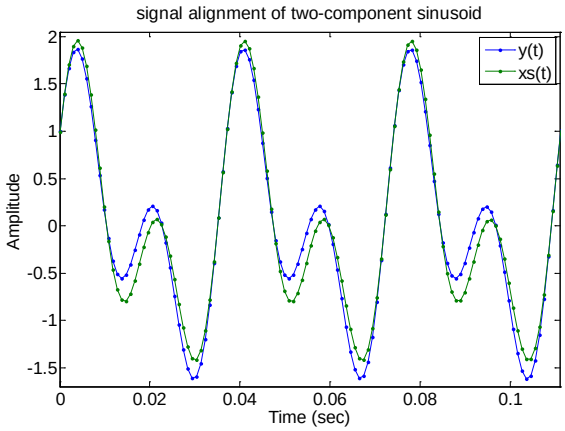


Figure 6. Simulated signal x_s & y showing that direct signal shifting would not work with uniform phase shift in multi-component sinusoid signal

2.2.2. Uniform time delay

For the process of synchronous signal averaging, any errors and/or differences in the phase reference signal and zero-crossing point will consequently produce a uniform time delay across all frequency components in the averaged signal. This is the underlying cause of phase misalignment between SSAs.

For the above example with

$$x(t) = \sin(2\pi \cdot 27 \cdot t + 0.987) + \sin(2\pi \cdot 2 \cdot 27 \cdot t + 1.053)$$

and

$$y(t) = x(t - 0.0205),$$

the time delay of 0.0205 seconds corresponds to almost 21 samples (i.e. 0.0205×1024 = 20.992), so direct signal shifting should work fine. However, direct signal shifting will not work when the uniform time delay is 0.02 seconds, as this corresponds to 20.48 samples. It would not be hard to imagine what difference this nearly half-a-sample shifting error is going to make in the change signal. This is a very likely scenario with synchronous signal averages because any differences between the phase reference signals from one signal average to the next are almost certainly going to occur in non-integer samples – although some can be really close to integers, such as the gear signals shown in section 2.1 where an optical shaft encoder was used.

An alternative approach to direct signal shifting in the time domain is to carry out the shift in the frequency domain. Figure 7 shows an example of aligning two signals involving a uniform time delay of 0.02 second (or 20.48 samples) by shifting the phase spectrum of x and then transforming back to the time domain. The theory behind this example will be given in the next section. We can see in Figure 7 that the shifted x (x_s) is perfectly aligned with signal y. In fact, this approach applies to both cases of uniform phase shift and uniform time delay.

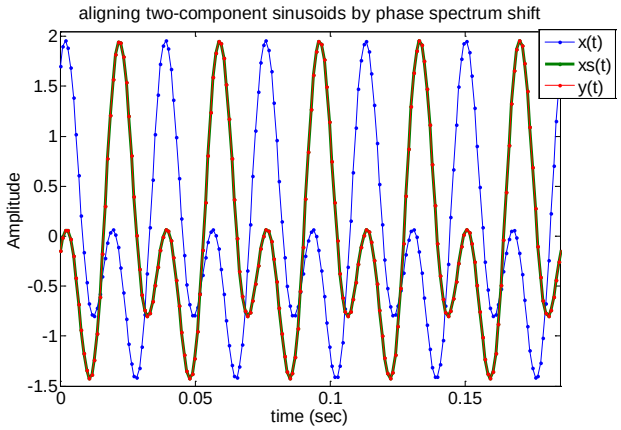


Figure 7. Alignment of signals with a 0.02 second time delay (at 1024 sampling rate) via shifting the phase spectrum of x by the difference between phase spectra of x and y

3. THEORETICAL DEVELOPMENT OF UNIFIED CHANGE DETECTION APPROACH

From the last section, we have shown that the future-state signal $y(t)$ can be aligned with the healthy-state signal $x(t)$ by introducing a time shift. In other words, alignment of the signals means a simple time shift by $-\Delta t$ which is the lag that gives the maximum value of the cross-correlation function. It can be carried out in the frequency domain. Mathematically, if we assume that signal $y(t)$ is a time shifted version of signal $x(t)$, and ignore the amplitude difference, we have

$$y(t) = x(t - \Delta t) \tag{1}$$

Taking the Fourier transform on both sides of Eq. (1) and making use of the translation property of Fourier transform, we get

$$\begin{aligned} Y(f) &= X(f)e^{-j2\pi\Delta t f} \\ A_Y(f)e^{j\Phi_Y(f)} &= A_X(f)e^{j\Phi_X(f)}e^{-j2\pi\Delta t f} \\ A_Y(f)e^{j\Phi_Y(f)} &= A_X(f)e^{j[\Phi_X(f)-2\pi\Delta t f]} \end{aligned} \tag{2}$$

where the amplitude and phase spectra are given by

$$\begin{aligned} A_X(f) &= |X(f)|, \quad \Phi_X(f) = \angle X(f) \\ A_Y(f) &= |Y(f)|, \quad \Phi_Y(f) = \angle Y(f) \end{aligned} \tag{3}$$

By ignoring the amplitude difference, or making $A_Y(f) = A_x(f)$, we have

$$\begin{aligned}\Phi_Y(f) &= \Phi_X(f) - 2\pi\,\Delta t\, f \\ 2\pi\,\Delta t\, f &= \Phi_X(f) - \Phi_Y(f)\end{aligned} \tag{4}$$

From Eqs. (2) and (4), we can see that time-shifting $x(t)$ by Δt , i.e. $x(t-\Delta t)$ is equivalent to shifting the phase spectrum of $x(t)$ by the difference of phase spectra of $x(t)$ and $y(t)$. The time-shifted $x(t)$ will be aligned with $y(t)$. Hence we don't really need to know the lag Δt via cross-correlation and interpolation.

Now, we put the amplitude difference back, the Fourier transforms of signal $x(t)$ and $y(t)$ are respectively given by

$$\begin{aligned}X(f) &= \int_{-\infty}^{\infty} x(t)\cdot e^{-j2\pi f t}\mathrm{d}t = A_X(f)\cdot e^{j\Phi_X(f)} \\ Y(f) &= \int_{-\infty}^{\infty} y(t)\cdot e^{-j2\pi f t}\mathrm{d}t = A_Y(f)\cdot e^{j\Phi_Y(f)}\end{aligned} \tag{5}$$

The difference of phase spectra is

$$\Delta\Phi_{XY}(f) = \Phi_X(f) - \Phi_Y(f) \tag{6}$$

Shifting the phase spectrum $X(f)$ by the difference given in Eq. (6), which is equivalent to time-shifting $x(t)$ by Δt , i.e. $x(t-\Delta t)$, we have the Fourier transform of the shifted signal

$$\begin{aligned}\hat{X}(f) &= A_X(f)e^{j[\Phi_X(f)-2\pi\Delta t f]} \\ &= A_X(f)\cdot e^{j\cdot[\Phi_X(f)-\Delta\Phi_{XY}(f)]} \\ &= A_X(f)\cdot e^{j\cdot[\Phi_X(f)-(\Phi_X(f)-\Phi_Y(f))]} \\ &= A_X(f)\cdot e^{j\cdot\Phi_Y(f)}\end{aligned} \tag{7}$$

We can derive the shifted version of signal $x(t)$ by an inverse Fourier transform

$$\hat{x}(t) = \int_{-\infty}^{\infty} \hat{X}(f)\cdot e^{j2\pi f t}\mathrm{d}f \tag{8}$$

which is a real-valued signal as $x(t)$ and $y(t)$ are both real-valued so that $A_X(f)$ is even and $\Phi_Y(f)$ is odd.

Having had $x(t)$ and $y(t)$ aligned, we can now define the change signal as

$$\delta_{xy}(t) = y(t) - \hat{x}(t) \tag{9}$$

On the other hand, we will see that the Fourier transform of the change signal is

$$\begin{aligned}\Delta_{XY}(f) &= Y(f) - \hat{X}(f) \\ &= A_Y(f)\cdot e^{j\cdot\Phi_Y(f)} - A_X(f)\cdot e^{j\cdot\Phi_Y(f)} \\ &= \left[A_Y(f) - A_X(f)\right]\cdot e^{j\cdot\Phi_Y(f)}\end{aligned} \tag{10}$$

Therefore, we can also define the change in the spectral domain. Notice that the amplitude in Eq. (10), $[A_Y(f)-A_X(f)]$, may be negative at some frequencies, which means a phase shift of π to those frequency components. The change signal in the time domain is then obtained by an inverse Fourier transform

$$\delta_{xy}(t) = \int_{-\infty}^{\infty} \Delta_{XY}(f)\cdot e^{j2\pi f t}\mathrm{d}f \tag{11}$$

For reasons mentioned in the first example in Section 2.1, it is often necessary to select a cross-over frequency in shaft orders to divide the change signal into high & low bands when changes are not obvious in the full-band. Therefore, for fault detection and trending purposes the change signal can be viewed from three perspectives, i.e. in low-band, high-band and full-band.

4. DERIVATION OF CONDITION INDICATORS

Three condition indicators (CIs) are defined in this section. These CIs can be used as measures of the machine health state; they can be trended over time for fault detection purposes. In Section 5, the unified change detection technique with these CIs is applied to the detection of several different types of faults.

4.1. Kurtosis of the change signal

Kurtosis is the 4th order statistical moment normalized by the standard deviation to the 4th power; it is often used as the CI for localized gear and bearing faults, such as gear tooth cracking and bearing element spalling. These local faults cause spikiness in fault signatures and kurtosis is an effective indicator for spikiness in the signal. For a discrete change signal $\delta(n)$, n = 1, 2, … N, with a mean value of $\bar{\delta}$, the kurtosis is defined as

$$K_\delta = \frac{N\left[\sum_{n=1}^{N}\left(\delta(n)-\bar{\delta}\right)^4\right]}{\left[\sum_{n=1}^{N}\left(\delta(n)-\bar{\delta}\right)^2\right]^2} \tag{12}$$

If the change signal is Gaussian noise, the above kurtosis will be around 3. In gear fault diagnosis, many healthy-state residual signals (after removing gear mesh harmonics and their sidebands) are sub-Gaussian with kurtosis values slightly less than 3. Kurtosis values of 3.5 and 4.5 are generally regarded as the alert and alarming levels respectively. Usually, the high-band kurtosis is more sensitive to sharp spikes induced by localized faults. However, kurtosis may not necessarily be good when it is used as a trending parameter because spikiness can be reduced in the change signal as localized fault develops into distributed fault, especially in cases of bearing faults.

4.2. Energy ratio of the change signal

The standard deviation or root mean square (RMS) value of the change signal can also be employed as a trending parameter to continuously monitor the condition changes in rotating machinery. We define energy ratio as the ratio between the RMS of the change signal and the RMS of the healthy-state or reference signal, i.e.

$$E_{\delta,x} = \frac{\sqrt{\sum_{n=1}^{N}\left(\delta(n)-\bar{\delta}\right)^2}}{\sqrt{\sum_{n=1}^{N}\left(x(n)-\bar{x}\right)^2}} \tag{13}$$

The energy ratio is used to normalize the energy in the change signal against the constant energy in the reference signal. Ideally, we can expect the energy ratio to increase as the fault progresses from early to late stages provided that the fault-induced changes are well reflected in the change signal. However, the randomness in the CI may not make the increasing trend strictly monotonic.

4.3. Scaled Kurtosis of the change signal

We define the scaled kurtosis as the product of the kurtosis of the change signal and the energy ratio given by Eqs. (12) & (13). Mathematically, the expression for scaled kurtosis is

$$\begin{aligned}\hat{K}_{\delta,x} &= K_{\delta}\cdot E_{\delta,x} \\ &= \frac{N\left[\sum_{n=1}^{N}\left(\delta(n)-\bar{\delta}\right)^4\right]}{\left[\sum_{n=1}^{N}\left(\delta(n)-\bar{\delta}\right)^2\right]^{3/2}\cdot\left[\sum_{n=1}^{N}\left(x(n)-\bar{x}\right)^2\right]^{1/2}}\end{aligned} \tag{14}$$

It combines the change signal with the reference signal, so that the condition is always compared to a common reference. As we can see in the following applications, this CI can give a more consistent trending of fault conditions than the kurtosis itself. A reasonable explanation for the results would be that, in the early stages of fault development (when the fault is localized), the kurtosis performs more effectively than the energy ratio. However, in late stages of fault development (when the fault may be more distributed), the spikiness in the change signal drops but the energy level in the change signal increases rapidly, which will lead to an overall increase in the scaled kurtosis (the product).

5. APPLICATIONS OF THE UNIFIED CHANGE DETECTION APPROACH

We have defined the approach to deriving the change signals from the healthy-state signal to the future-state signals. With the change signals, we have proposed three condition indicators in three frequency bands. This will produce nine CIs for each future-state signal. Trending these CIs over time will allow changes in the condition of the monitored component of the machine to be detected. In this section, we will demonstrate the effectiveness and robustness of the proposed method in a number of different fault cases involving different fault types.

Vibration data from machines with several types of faults are used for the demonstration. The fault types include gear tooth cracks in a simple gearbox; non-uniform gear tooth wear and vane pump failure in turbo-machinery; and nut looseness and planet carrier plate cracking in helicopter transmission systems. Using the same unified approach, we have produced various trending curves for each of these fault types. The results have shown that this single unified change detection approach can be very powerful in detecting changes caused by many different types of machine faults. In practice all nine CIs should be trended during machine operations. As there is not enough space in this paper to show results for all nine CIs, we will show results for some selected CIs in the following examples.

5.1. Application to Detecting Gear Tooth Crack Growth

The study of tooth crack development and propagation in the pinion spur gear of a test gearbox were performed by Swinburne University of Technology and DSTO (Forrester 1996, Vavlitis 1998). The test gearbox was a simple single-stage reduction gearbox with 27 teeth on the driving pinion and 49 teeth on the driven gear (i.e. the gear ratio was r = 27/49). The gearbox was driven by an electric motor through a belt drive. The load to the test gear was provided by a dynamometer with a full loading capacity of 45kW at 40Hz input shaft speed. The test gears, labeled G6, A1, A2, A3 and A5 etc, were the input pinion (with a rated load of 27.5kW) with a semi-circular spark-eroded notch (2mm×0.1mm×1mm) at the root fillet in the middle of the tooth width. The notch was designed as a stress riser for crack initiation during the test. The gear was made of EN36A case hardening low alloy steel with teeth precision-ground under AGMA Class 13 standard. The input speed of the gearbox was set to a nominal value of 2400rpm (40Hz), which was varied during the test in a range of 38.6 to 39.3Hz. Results with selected CIs for G6, A3 & A5 are shown here.

Figure 8 shows the trending curve for G6 scaled kurtosis in the high band (cross-over at 85th shaft order) from files G6b.071 to G6b.110, the dataset used here is the same as that used in Section 2.1. We can see that the scaled kurtosis CI generally trends upwards with the increasing crack size. However, the general trend was disrupted at file #97. This was caused by the inspection after file #96, where the faulty gear was dismantled from the test rig and the tooth crack was forced to open with a static overload for magnetic rubber inspection of crack size. It is believed that the inspection process interrupted the crack progression, i.e. the static overload caused crack retardation or arrest. Figure 9

shows the trending curves of energy ratio CIs in full, high and low order bands. Obviously, the high band (blue line) is most sensitive to the changes caused the increasing crack size.

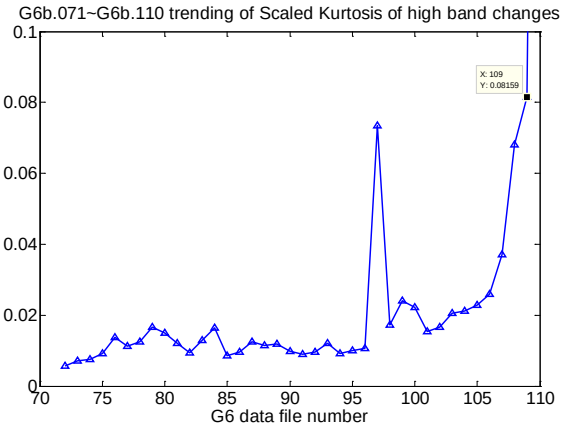


Figure 8. Trending of G6 scaled kurtosis in high-band from G6b.071 to G6b.110 (the value at G6b.110 was 0.66 – outside the displayed range).

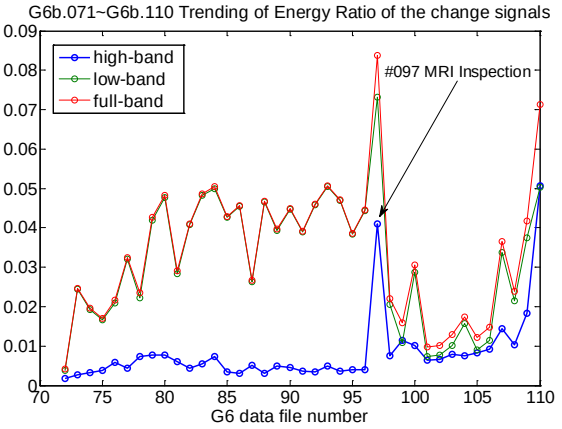


Figure 9. Trending of G6 energy ratio in all three bands for (G6b.071~110) under 45kW load, crack growth disrupted by an inspection.

While the results shown in Figure 8 and Figure 9 are possibly affected by interrupted tooth crack growth, Figure 10 shows the trending curve of high band kurtosis for an uninterrupted growth from G6b.149 to G6b.155. The reference signal was G6b.148 where the tooth crack size was estimated to be 3.63mm by post-test fractography analysis. By G6b.155 (the last data file for the G6 test), the tooth crack grew to an advanced stage where the cracked tooth was just about to fall off, and the crack length was measured at 4.67mm by fractography analysis (80 percent tooth body cracked, as compared to the crack path length of 5.82mm). Note that the kurtosis values in this plot do not represent the change between the faulty-state and healthy-state, rather the change was from a 'less faulty' to 'more faulty' state (i.e. the normal alert and alarm levels of 3.5 and 4.5 do not apply here). Figure 11 shows the change signals from G6b.148 to G6b.155 in the high and low bands.

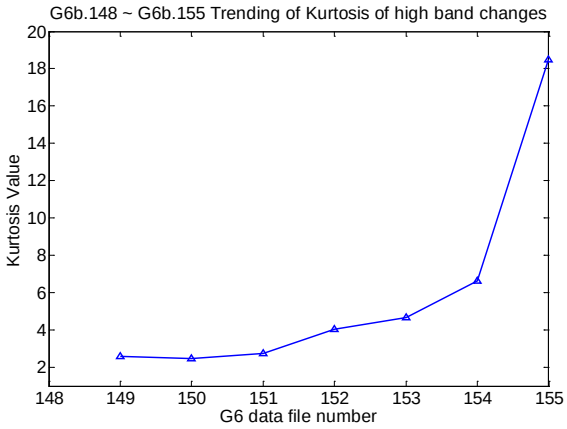


Figure 10. Trending of G6 kurtosis in high-band for G6b.148~155, a further uninterrupted crack growth under constant load (24.5kW) after G6b.110.

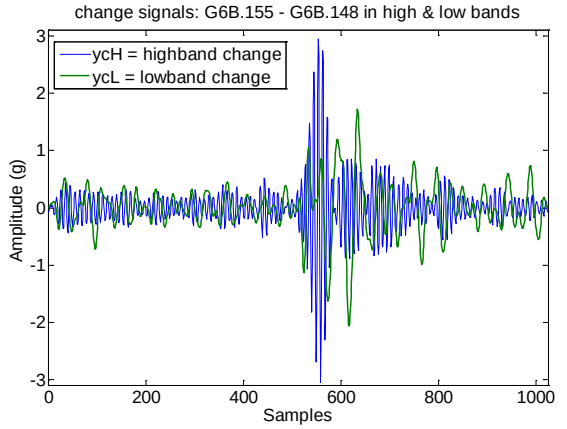


Figure 11. Change signal from G6b.148 to G6b.155.

More results are given in Figure 12 and Figure 13 for the DSTO gear tooth crack propagation test series using the identical type of test gears. Figure 12 shows the A3 gear test trending curve of scaled kurtosis from A3B2.501 to A3B2.549 over some 27.5 minutes of testing (about 66000 fatigue cycles to the cracked tooth with constant load of 30kW at 40Hz shaft speed). In this test period, the crack had uninterrupted continuous growth from 4.89mm to 5.84mm along a curved crack path (Vavlitis 1998, where A3 was labelled as A2-3). With file A3B2.549, the kurtosis of the change signal in high order band is 9.49, as compared to the conventional residual kurtosis of 5.09.

Similarly, Figure 13 shows the A5 gear test trending curve of scaled kurtosis from A5B0.598 to A5B0.763 over some 84 minutes of testing (about 201600 fatigue cycles to the cracked tooth with 40Hz shaft speed) where the crack had uninterrupted continuous growth from 1.46mm to 2.27mm along a curved crack path (Vavlitis 1998, where A5 was

labelled as A2-5). With file A5B0.763, the kurtosis of the change signal in the high order band is 9.0, as compared to the conventional residual kurtosis of 4.3. If we pay close attention to the values on the vertical coordinate (scaled-kurtosis) in Figure 12 and Figure 13, we could find that these values might be a reflection of the crack sizes, e.g. the scaled kurtosis value of 0.44 for A3B2.549 with a crack length of 5.84mm versus the scaled kurtosis value of 0.174 for A5B0.763 with a crack length of 2.27mm. However, this could also be affected by the load.

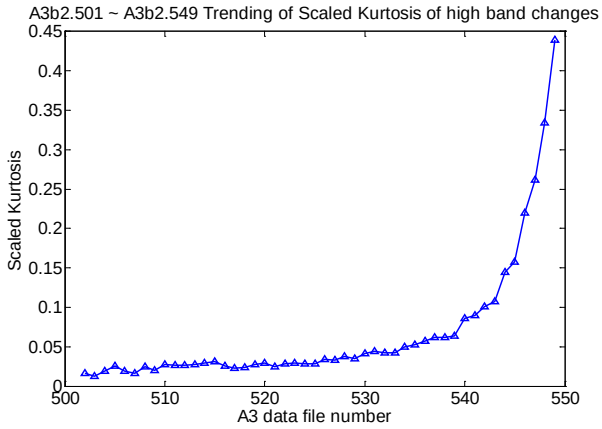


Figure 12. CI trending of A3B2.501~549 data – final crack size 5.84mm with uninterrupted crack growth under 30kW load.

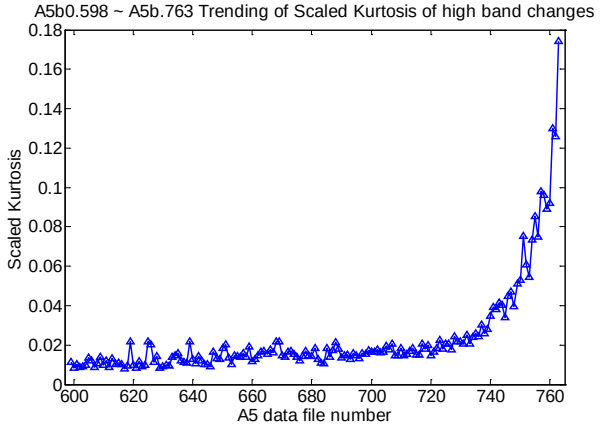


Figure 13. CI trending of A5B0.598~763 data – final crack size 2.27mm with continuous progression without interruption under constant load of 45kW.

We have also conducted a more detailed comparison study between the unified change detection approach and other commonly used gear fault detection techniques. Figure 14 shows the results of comparing the unified approach with two other methods based on the autoregressive (AR) model residual and the conventional residual signals using the A3 gear test data. We find that the changes picked up by the unified approach increase more rapidly than the other two methods, and the AR model result is very much dependent on the selection of model order, and whether the AR model is built on a reference signal or the monitored signal itself. The unified approach has shown more fluctuation in the result, which could be smoothed out by using the scaled kurtosis as shown in Figure 12.

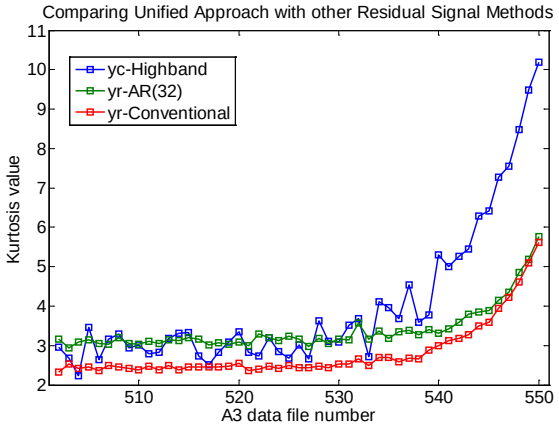


Figure 14. Comparative study between unified change detection approach and other methods based on self-AR(32) residual and conventional residual signals using the DSTO A3 gear tooth cracking data

We can draw some conclusions based on the results of the comparison study using the DSTO gear rig data. The unified approach: (1) requires less prior knowledge, it only needs to choose the high & low band cross-over frequency (e.g. at the lower bound of a resonance or the upper bound of the significant gear mesh harmonics); (2) is much more versatile than conventional residual signal method in which we must know which orders to be removed; (3) is capable of dealing with cases of multiple gears on the same shaft and is more robust than the AR residual method where a consistent model order selection is lacking; and (4) gives better and more robust trending capability by using a scaled kurtosis CI than a conventional kurtosis CI.

5.2. Application to Detecting Faults in Turbomachinery

We have found in the last section that the unified change detection approach is effective in detecting localized changes induced by gear tooth cracking, especially by using a high band CI. In this section, we will find if this approach can be employed for the detection of distributed faults such as uneven wear on many teeth of a gear, and damage to all the vanes of a vane pump. The results show that low-band and full-band CIs are very sensitive to the changes caused by these distributed faults.

5.2.1. Non-uniform Gear Tooth Wear

In gear design, it is normal practice to select the number of teeth for a gear pair such that there is no common factor between them. This allows each tooth of one gear to mesh with every tooth of the other gear, and therefore promotes

even wear of the teeth. This system is usually referred to as the hunting tooth system. In the gearbox of a developmental turbofan, there was a non-hunting tooth system with a common factor of 3 between the tooth counts, which resulted in damage (non localized uneven wear) to every 3rd tooth on the pinion. Now, we employ the unified change detection approach to monitor the changes induced by this specific wear pattern over time.

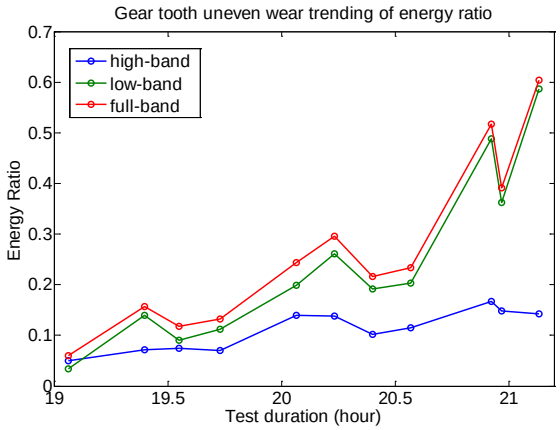


Figure 15. Turbo-fan gear CI (energy ratio) trending with Channel 3 data and cross-over frequency of 5 shaft orders.

The reference signal *x* was acquired at about 19 hours of testing, and 12 monitoring signals (*y*) were acquired after signal *x*. Figure 15 shows the trending curve of the energy ratio CI in full, low and high order bands versus the accelerated test time over a period of more than 2 hours. The increasing trend of the full (red line) and low (green) band CIs clearly shows the progression of the uneven wear to every 3rd tooth on the pinion gear. The high (blue) band CI has a less obvious trend. These results were obtained with a cross-over frequency of 5 shaft orders; so it means that most of the energy in the change signal is in the low band below the 5th shaft order. In fact, the high band CI is of little importance in this case as the distributed fault would not necessarily bring any high frequency resonance features.

5.2.2. Vane Pump Failure

This fault type is about the severe damage to all the vanes in a vane pump attached to the accessory gearbox of an aircraft engine. The vibration data were recorded at three stages of an accelerated test when the engine was running on full power. They were from *a*) early stage – within the first 10 percent of the testing time; *b*) late stage – between 80 ~ 90 percent of the testing time and *c*) last stage – within the last 2 percent of the testing time of the accelerated test. Altogether, there were 36 tri-axial vibration data files used for producing the results shown in this paper, where the first one in the early stage was used as the reference.

Figure 16 shows the trending curves of scaled kurtosis CI in three bands using the horizontal axial (the most sensitive direction) vibration data. Along the abscissa coordinate of the plot there are 35 columns of CI points; the first 11 files were from the early stage of testing, the following 18 files were from the late stage and the last 6 files from the last stage of testing. The cross-over frequency for the low and high bands was selected at just above the 6th harmonic of the vane pass frequency. We can see in Figure 16 that the full band (red) and low band (green) CIs show prominent step changes across the three stages of testing. The high band (blue) CIs show some indication of change but this is not as prominent as the other two bands. This is because the signal changes caused by the vane damage are mostly likely located at the vane pass frequency and its lower harmonics. Obviously, the changes detected by the unified approach can give sufficient lead time to the failure of the vane pump. The pump actually failed on the very next run after the last data file was recorded.

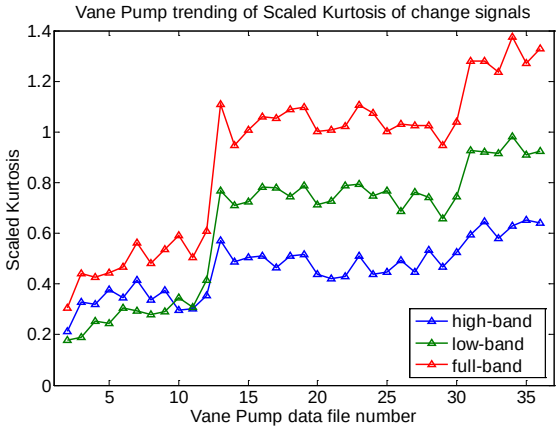


Figure 16. Aero-engine vane pump CI (scaled-kurtosis) trending with Channel 3 data and cross-over frequency of 65 shaft orders.

5.3. Applications to Detecting Faults in Helicopter Transmission Systems

Health and Usage Monitoring Systems (HUMS) have been used in helicopter transmission gearboxes for many years. In general, existing HUMS can detect faults of common types such as gear and bearing faults without great difficulty. However, less common or unknown types of faults are difficult to detect. In this section, we will present two cases of less common fault types and employ the proposed unified approach to trend the progression of these faults.

5.3.1. Input Shaft End Nut Looseness

The first of these less common fault types is the end nut looseness at the bevel input pinion extension shaft in a helicopter Main Rotor Gearbox (MRG). This is a fault type which is believed to be the most likely cause of the rupture

of the extension shaft. It can be induced by a lack of tightening torque of the end-nut and consequently causes a load redistribution in the MRG assembly. A study was conducted at DSTO into this fault type using a light utility helicopter MRG in DSTO's Helicopter Transmission Test Facility (HTTF). The objective of the study was to provide HUMS systems with the capability to detect the loss of tightening torque of the end-nut and to prevent the rupture of the input pinion extension shaft.

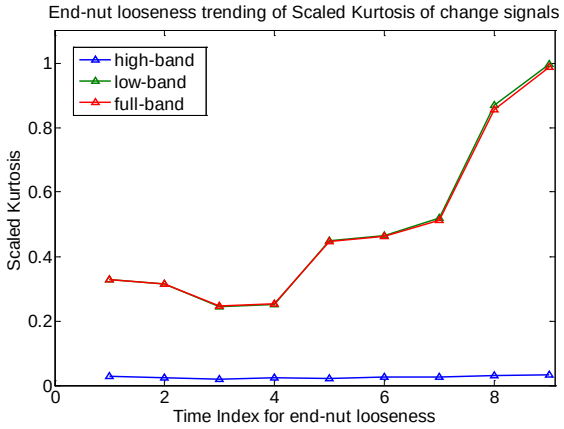


Figure 17. Trending of scaled kurtosis CI from pinion SSA (at Ring-Front sensor & cruise power) change signal with cross-over at 75th shaft order

Ten end-nut tightening torques were used in the test, i.e., 100 percent, 91%, 78%, 67%, 56%, 44%, 33%, 22%, 11% and 7% of the nominal tightening torque. The data recorded at 100% tightening torque were used as the reference, and the tightening torque is assumed to become less and less over time. The 7 percent torque (a very loose condition) was found to be the thread breaking torque at which we could just start to turn the end-nut. Throughout the test, the input shaft speed was kept at the nominal level (about 100Hz) and there was no mast load applied to the MRG. The data used in this paper were acquired under the forward flight condition at 75 percent maximum power.

Using the synchronous signal averages (SSA) with respect to the input pinion shaft and the planet carrier shaft, we produced the scaled-kurtosis CIs at each level of the tightening torque and plotted them in Figure 17 and Figure 18. The abscissa coordinates in the plots can be considered a time progression index where each point corresponds to the next looser level of the tightening torque, i.e. time index 1 corresponds to the 91%, index 2 is 78% … and index 9 is 7% tightening torque.

From Figure 17 which is based on the input pinion SSA change signals, we can see that the end-nut loosening condition can be detected by the full (red) and low band (green) CIs from time index #5 (i.e. 44% tightening torque), and becomes very obvious at index #8 (or 11% tightening torque). On the other hand, Figure 18 shows the CI trending based on the planet carrier SSA change signals. Here, it could be argued that the end-nut loosening condition is detectable by the full (red) and high band (blue) CIs from time index #3 (i.e. 67% tightening torque) forward, which is apparently better than the result in Figure 17. This result may be because the effect of load redistribution caused by the loosening end-nut on the input shaft was magnified at the carrier shaft by the reduced speed and increased torque. The results have shown that the unified approach can be effective in detecting faults of this particular type.

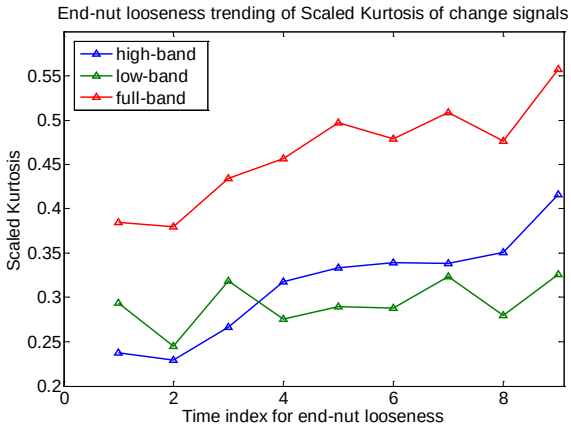


Figure 18. Trending of scaled kurtosis CI from carrier SSA (at Ring-Front sensor & cruise power) change signal with cross-over at 750th shaft order

5.3.2. Planet Carrier Plate Cracking

The helicopter main gearbox planet carrier plate cracking was not a widely known fault type until 2002 when it occurred in the UH-60A Blackhawks of US Army. Since 2004, it has also occurred in the SH-60B Seahawks of US Navy. The test data used for this paper were acquired at US Navy's HTTF in Patuxent River, Maryland.

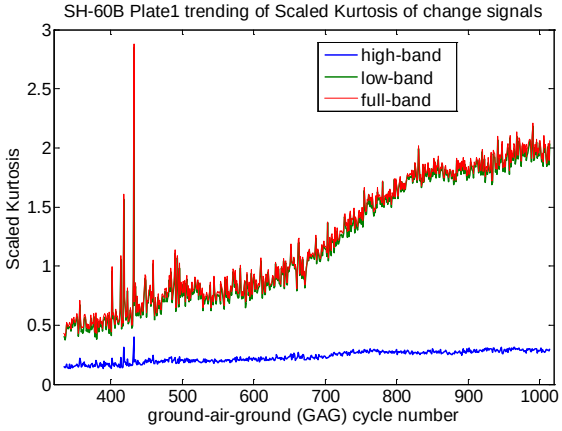


Figure 19. Trending of scaled kurtosis CI of 40% torque and STBDRING sensor at cross-over of 1700 shaft order.

Using the unified approach, we produced CIs for all the sensor data. Some results with selected HUMS sensors at 40 percent torque for the main rotor are shown in Figure 19 to Figure 21. With a cross-over frequency of 1700 orders of the carrier shaft, and vibration data from the sensor on the starboard side of the ring gear (STBDRING), the scaled kurtosis CIs versus ground-air-ground (GAG) cycle number (equivalent to a time index) are shown in Figure 19. We can see that the full (red) and low (green) band CIs track well with the changes caused by the crack propagation in which the crack lengths were known to have grown from 90mm (3.54") at GAG #410 to 172mm (6.78") at GAG #763.

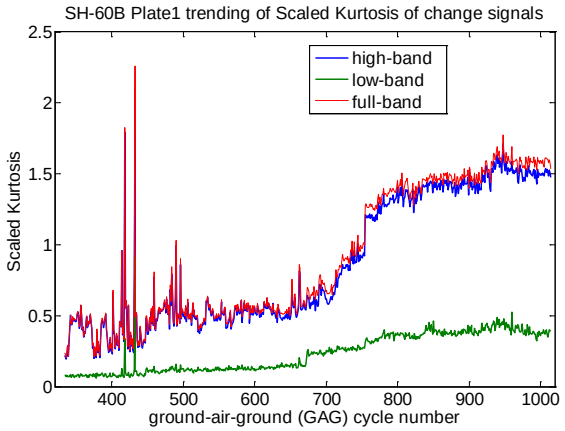


Figure 20. Trending of scaled kurtosis CI of 40% torque and VMEP1 sensor at cross-over of 500 shaft order.

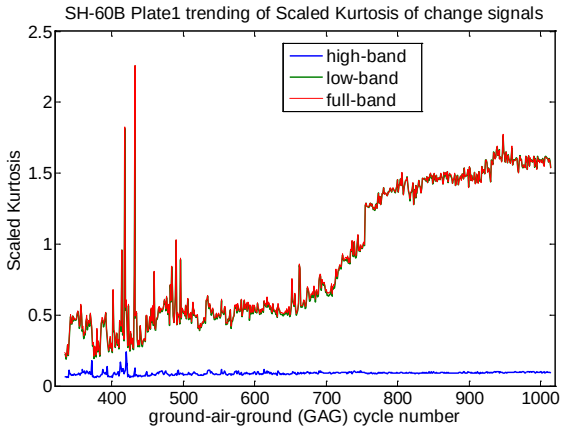


Figure 21. Trending of scaled kurtosis CI of 40% torque and VMEP1 sensor at cross-over of 1700 shaft order.

Figure 20 and Figure 21 show the results for another sensor (VMEP1, which was very close to the STBDRING sensor) at 40% main rotor torque with two different cross-over orders, i.e. 500 and 1700 orders, respectively, to show the effect of cross-over frequency on the fault detectability of the unified approach. Note that the ring gear has 228 teeth so 500 shaft orders is above the 2nd gear meshing harmonic, and there were no significant meshing harmonics beyond 1700 shaft orders. Obviously, the full band (red) CIs are identical in the two plots, which track well with the crack growth. In particular, the CI had a sudden jump at GAG #755 where the crack propagated through the outer edge of the carrier plate, which was not evident in Figure 19. Interestingly, the high band CI (blue) in Figure 20 and the low band CI (green) in Figure 21 are almost identical to the full band CIs. This means that the energy in the change signals is concentrated between 500 and 1700 orders of the carrier shaft.

Based on the results in Figure 20 and Figure 21, we can say that the selection of cross-over frequency (or order) doesn't affect fault detectability of the unified approach as a whole; it can however provide further diagnostic information on where the energy in the change signal is located in the frequency domain. The energy bandwidth in the change signals may well be utilized to distinguish the localized faults (with high bandwidth features) from the distributed faults (with low bandwidth features). We need to notice that, in this example, the cross-over orders of 500 and 1700 correspond to the frequencies of 2150Hz and 7310Hz, i.e. the order times the main rotor speed of 4.3Hz.

6. DISCUSSION AND CONCLUSIONS

In this paper we have presented a unified change detection approach to generalized health monitoring for rotating machinery. The approach is based on aligning the signals through shifting the phase spectrum of the healthy-state or reference signal by the difference in phase spectra from the future-state signal (or the signal under monitoring). The change signals are obtained from direct subtraction of the aligned signals. Condition indicators extracted from the change signals are used to detect changes caused by machine faults. Results have shown that the proposed unified approach is very effective and robust in detecting changes caused by various types of mechanical faults.

In practice, failure modes sometimes occur which were not anticipated in the development of a machine condition monitoring system, and these can often remain undetected, with potentially catastrophic consequences. It is unfortunate that we are unable to detect such faults as they happen and must come up with new techniques to detect them when they occur again. It has been our intention to develop a powerful unified fault detection method to deal with new or unexpected failure modes (or types of faults) in rotating machines. The proposed method has provided some hope in achieving that goal.

Threshold setting for the CIs is a very important aspect in HUMS and PHM systems. The kurtosis can have a threshold of 3.5 for reasons mentioned in Sections 2.1 and 4.1. The energy ratio should certainly have a threshold below 1 based on its definition in Eq. (13); hence a reasonable one would be 0.5 – meaning that the energy in

the change signal has reached 50 percent of that in the reference signal. However, there is no common threshold of scaled kurtosis for different fault types as observed in the results of this paper. From the definition of the scaled kurtosis in Eq. (14), we may look at setting the threshold upper bound to around 1, e.g. an energy ratio of 0.33 and a kurtosis of 3 ($0.33\times3\approx1$). Another way of thresholding the scaled kurtosis may be to put a limit on its rate of change (or differential). This will be an area for further study.

The other area worth further investigation is the systematic approach to selecting the reference signal. Is it always sufficient to just use the data at the beginning of the machine operation, or is it better to choose the data at the start of each run or use a moving reference signal? These are questions to be answered after further testing and validating of the proposed unified approach against a wide range of fault types.

In conclusion, we have shown that the proposed unified change detection approach is effective and robust in detecting changes caused many types of mechanical faults. It has the potential to cope with a much wider range of failure modes in rotating machinery than the existing methods. The new method is also simple in concept and fast in calculation (it only needs the FFT), and would be straightforward to implement in existing PHM systems. We anticipate that the method could be widely tested and matured in the near future.

ACKNOWLEDGEMENT

The authors would like to acknowledge Dr. David Blunt of DSTO for his review of the paper. We also give acknowledgements to JSF Program Office, US Navy, Pratt & Whitney, Rolls-Royce Corporation and Eurocopter for their collaborations.

REFERENCES

Becker, A., Peters, W. and Konzen, M. (2007). Advanced Vibration Monitoring of Sea King Input Shaft Assembly. *International Conference of Maintenance Societies*, May 2007, Melbourne, Australia.

Blunt, D.M and Keller, J.A. (2006). Detection of a fatigue crack in a UH-60A planet gear carrier using vibration analysis. *Journal of Mechanical Systems and Signal Processing 20*(8), pages 2095-2111.

Bonnardot, F., Badaoui, M.El, Randall, R.B., Daniere, J., and Guillet, F. (2005). Use of the acceleration signal of a gearbox in order to perform angular resampling (with limited speed fluctuation). *Journal of Mechanical Systems and Signal Processing 19*(6), pages 766-785.

Braun, S. (1975). Extraction of Periodic Waveforms by Time Domain Averaging. *Acoustica* 32, pages 69-77, 1975.

Burchill, R.F., Frarey, J.L. and Wilson, D.S. (1973). New machinery health diagnostic techniques using high-frequency vibration. *SAE Paper 730930*, 1973.

Forrester, B.D. (1996). *Advanced Vibration Analysis Techniques for Fault Detection and Diagnosis in Geared Transmission Systems*. Ph.D. Thesis, Swinburne University of Technology, Australia.

Galati, F.A., Forrester, B.D. and Dey, S. (2008). Application of the generalised likelihood ratio algorithm to the detection of a bearing fault in a helicopter transmission. *Australian Journal of Mechanical Engineering* 5(2), pages 169-176, 2008.

Galati, F.A. (2007). Investigation into Input Drive-Shaft Assembly Failures in RAN Sea King's. *Vertiflite Conference (AIAC12)*, 13-17 March 2007, Melbourne, Australia.

Hancock, K.M. and Zhang, Q. (2006). A Hybrid Approach to Hydraulic Vane Pump Condition Monitoring and Fault Detection. *Transaction of the American Society of Agricultural and Biological Engineers (ASABE)* 49(4): 1203−1211, 2006.

Larder, B.D. (1999). Helicopter HUM/FDR: Benefits and Developments. *Proceedings of the 55th Annual Forum of the American Helicopter Society (AHS)*, 25-27 May 1999, Montreal, Canada.

Jarvis, M.P. and Sleight, P. (2011). *Report on the Accident to Aerospatiale (Eurocopter) AS332 L2 Super Puma Registration G-REDL*. Aircraft Accident Report 2/ 2011, Air Accidents Investigation Branch, Department of Transport, UK.

Lee, E. (2010). A simple HUMS approach to detect characteristic variation for mechanical systems. *Prognostics and Health Management Conference* (PHM '10, p1-8), 12-14 Jan 2010, Macau, China.

Lei, Y.G., Lin, J., He, Z.J., and Zuo, M.J. (2013). A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Journal of Mechanical Systems and Signal Processing* 35, pp. 108-126, 2013.

Man, Z.H., Wang, W., Khoo, S.Y., and Yin, J.L. (2012). Optimal Sinusoidal Modelling of Gear Mesh Vibration Signals for Gear Diagnosis and Prognosis. *Journal of Mechanical Systems and Signal Processing* 33(11), pp. 256-274, Nov. 2012.

Mcfadden, P.D. and Toozhy, M.M. (2000). Application of Synchronous Averaging to Vibration Monitoring of Rolling Element Bearings. *Journal of Mechanical Systems and Signal Processing* 14(6), pp. 891-906, June 2000.

Randall, R.B. and Antoni, J. (2011). Rolling element bearing diagnostics – A Tutorial. *Journal of Mechanical Systems and Signal Processing* 25, pp. 485-520, 2011.

Stewart, R.M. (1977). Some useful data analysis techniques for gearbox diagnostics. *Technical report paper MHM/R/10/77*, University of Southampton, Institute of Sound and Vibration Research, 1977.

Vavlitis, C. (1998). Crack Growth Behaviour of Spur Gears: A Fractographic Analysis. *DSTO-TN-0137*, Defence Science and Technology Organisation, Jan 1998, Australia.

Vecer, P., Kreidl, M. and Smid, R. (2005). Condition Indicators for Gearbox Condition Monitoring Systems. *Acta Polytechnica Report* Vol. 45, No. 6/2005, Czech Technical University Publishing House, Prague, Czech Republic.

Wang, W.Y. and Harrap, M.J. (1996). Condition Monitoring of Ball Bearings Using Envelope Autocorrelation Technique. *The International Journal of Machine Vibration* (5), pages 34-44, 1996.

Wang, W. and Wong, A.K. (2000). Linear Prediction and Gear Fault Diagnosis. *Proceedings of the 13th International Conference of Condition Monitoring and Diagnostic Engineering Management (COMADEM)* 2000, pages 305-315, December 3-8, 2000, Houston, Texas, USA.

Wang, W. (2001). Early Detection of Gear Tooth Cracking Using the Resonance Demodulation Technique. *Mechanical System and Signal processing* 15(5), pages 887-903.

Wang, W. and Keller, J.A. (2007). A Novel Technique of Crack Detection for Helicopter Main Gearbox Planet Carrier. *Proceedings of the 5th DSTO International Conference on Health and Usage Monitoring* (HUMS 2007), 13-17 March 2007, Melbourne, Australia.

Wang, W. (2008). Autoregressive Model-Based Diagnostics for Gears and Bearings. *Insight – Non-Destructive Testing and Condition Monitoring* 50(8), pages 414-418, August 2008.

Wang, W. and Muschlitz, G. (2010). Disk Crack Detection in Spin Testing Using Tip Timing Data. *31st IEEE Aerospace Conference*, March 6-13 2010, Big Sky, Montana, USA.

BIOGRAPHIES

Wenyi Wang received his PhD in Mechanical Engineering from the University of New South Wales (UNSW) and became a Lecturer at Monash University in 1996. He joined DSTO in 1998 and has been a Senior Scientist in machine dynamics and diagnostics since 2002. Wenyi has over 60 publications and has received some prestigious awards including a Defence Science Fellowship in 2007 and a Victoria Fellowship in 2001. He is currently leading a JSF Program Office funded project on advanced vibration-based PHM for JSF propulsion systems.

B. David Forrester received his PhD in Mechanical Engineering from Swinburne University of Technology in 1996. He joined DSTO in 1981, and has been doing research in machine condition monitoring and diagnostics since 1988. He has been Principal Scientist and Head of machine dynamics and diagnostics at DSTO since 2010.

Peter Frith received his PhD in Mechanical Engineering from the University of New South Wales in 1984. He joined DSTO in 1983 and his research has focused on the development of health management technologies for propulsion and power systems of military aircraft. Peter was posted to the JSF Program Office from 2003 to 2006, and he was Scientific Adviser to the Royal Australian Air Force from 2008 to 2011. He is currently the Research Leader of the Aircraft Health and Sustainment Branch in the Aerospace Division of DSTO.

Evolving Fuzzy Classifier based on Clustering Algorithm and Drift Detection for Fault Diagnosis Applications

Maurilio Inacio[1], Andre Lemos[2], and Walmir Caminhas[3]

[1] *Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, 31270-901, Brazil*

[1] *Dept. of Computer Engineering, Faculdade de Ciencia e Tecnologia de Montes Claros, Montes Claros, 39400-142, Brazil*
maurilio.j.inacio@gmail.com

[2,3] *Dept. of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, 31270-901, Brazil*
andrepaim@gmail.com
caminhas@cpdee.ufmg.br

ABSTRACT

Nowadays, in several areas, efficient fault diagnosis methods for complex machinery and equipments are required. Several fault diagnosis methods based on different theories and approaches have been proposed in the literature. In general, these methods use mathematical/statistical models, accumulated experience, or even process historical data to perform fault diagnosis. Although methods based on models or experience have shown to be effective, they have the disadvantage of requiring previous knowledge of the dynamic system in question. On the contrary, methods based on process historical data do not require a *prior* knowledge, they are based solely on data obtained directly from the dynamic system. The application of so-called "Evolving Intelligent Systems" to accomplish fault diagnosis from process data have been shown a promising approach. This paper proposes an evolving fuzzy classifier based on a new approach that combines a recursive clustering algorithm and a drift detection method and its application on dynamic systems fault diagnosis. The novel approach provides greater robustness to outliers and noise present in data from process sensors. The classifier is evaluated in fault diagnosis of an interacting tank system and the results are promising.

1. INTRODUCTION

Nowadays, the advance of technology has resulted in the emergence of machinery and complex equipments, which impose great challenges for its management and maintenance. In many industries, for instance, fault diagnosis in major processes is vitally important to assure normal operation of a plant and avoid economic losses, security reductions and environmental damages. This context led to the emergence of new concepts on management and maintenance of machinery and equipments, such as Condition-Based Maintenance (CBM). In CBM, machine or equipment data obtained in real time are used to infer its working condition (or faulty condition), allowing maintenance scheduling and preventing equipment crashes. Another concept has emerged based on CBM, the concept of intelligent maintenance (Vachtsevanos, Lewis, Roeme, Hess, & Wu, 2006).

In past decades several fault diagnosis methods based on different approaches have been proposed in the literature. These methods use mathematical models, statistical models, accumulated experience, or process historical data to perform fault diagnosis (Venkatasubramanian, 2005). Fault diagnosis methods based on process historical data have received great emphasis recently (Abellan-Nebot & Subirón, 2010) and several works have already proposed data based diagnostics methods employing intelligent systems, mainly artificial neural networks and fuzzy systems (Jardine, Lin, & Banjevic, 2006). Nevertheless, despite the good performance achieved by intelligent systems in fault diagnosis, they tend to face difficulties when the problem involves complex non-stationary dynamic systems. In this systems, physical parameters, operating characteristics and fault behaviours change over time, requiring an adaptive fault diagnosis system, able to self-adapt to cope with changes in the monitored system. In order to address fault diagnosis in this cases, some works propose the use of so-called "Evolving Intelligent Systems" (Lughofer & Guardiola, 2008; Filev, Chinnam, Tseng, & Baruah, 2010; Lemos, Caminhas, & Gomide, 2013).

Based on artificial neural networks, fuzzy inference systems or a combination of both, the neurofuzzy networks, the evolv-

Maurilio Inacio et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ing intelligent systems are systems whose main characteristic is the ability to gradually determine both its structure and parameters from input data acquired in online mode and often in real time. Evolving intelligent systems applications has been growing in recent years. Many authors have obtained successful applications in real world complex problems involving modeling, control, classification or prediction (Angelov, Filev, & Kasabov, 2010). Evolving clustering algorithm is the most widely used approach to define the structure of an evolving intelligent system (Kasabov & Song, 2002; Angelov & Filev, 2003; Leng, McGinnity, & Prasad, 2005; Rong, Sundararajan, Huang, & Saratchandran, 2006; Lughofer, 2008; Soleimani-B., Lucas, & Araabi, 2010; Lima, Hell, Gomide, & Ballini, 2010; Lemos, Caminhas, & Gomide, 2011). This algorithms generally adopt a mechanism to update the structure (creation/modification/removal of clusters) and parameters of the system using some measure of similarity between input data samples and existing clusters. This mechanism may lead to an erroneous definition of the structure, since outliers or noisy samples (as usually are the data acquired by sensors in industrial environments) which exceeds the measure of similarity can generate clusters that do not effectively represent the data spacial structure (Lemos et al., 2011).

In fault diagnosis problems, the use of evolving intelligent systems based on recursive clustering algorithms robust to outliers and data noise is mandatory. In this case, each new cluster created is usually associated with a new faulty condition. Thus, if the clustering procedure is not robust, the fault diagnosis model tends to have a high false alarm rate, i.e., new faulty conditions are erroneously detected. Considering this context, this paper proposes a fault diagnosis approach based on an evolving fuzzy classifier which uses a new robust unsupervised recursive clustering algorithm. The unsupervised recursive clustering algorithm classifier consists of a modified version of the Gustafson-Kessel (GK) clustering algorithm (Gustafson & Kessel, 1979) with the incorporation of the Drift Detection Method (DDM) (Gama, Medas, Castillo, & Rodrigues, 2004).

Considered a powerful clustering algorithm, GK clustering algorithm unlike many others allows the identification of clusters with different shapes and orientations in space. The algorithm employs a technique to adapt the distance metric to the shape of each cluster using a estimation of the cluster covariance matrix. Furthermore, the algorithm has also the advantage of being relatively insensitive to data scale and initialization of the partition matrix (Filev & Georgieva, 2010). Drift detection, according to the literature, is a method to detect gradual changes in the context of input data. By context, it is understood as a set of generated data when the process is stationary. Drift detection methods are suitable for applications involving machine learning, where algorithms are applied to real world problems, in complex, non-stationary and dynamic environments (Sebastião & Gama, 2009). Among several methods proposed for drift detection, the DDM algorithm employs simple and computationally efficient method to detect moments when changes occur and it can be embedded into any learning algorithm, increasing its efficiency in problems involving non-stationary dynamic models.

In this paper, a new unsupervised recursive clustering algorithm is proposed, where any clustering update depends not only on the similarity measure, but also on the monitoring changes in the input data flow, which gives the algorithm a greater robustness to the presence of outliers and noise. A merging cluster mechanism was also incorporated into the algorithm to enable the removal of redundant clusters. The fuzzy rule base of the proposed classifier is updated whenever the cluster structure is modified. The clusters centers and covariance matrices are used as parameters of fuzzy rules. Multivariate Gaussian membership functions are employed in the rules to avoid information loss when there is interaction between input variables. Regarding the characteristics of the proposed recursive clustering algorithm, the main benefits achieved by the classifier used in this work are: 1) the ability to learn the dynamic system model in online mode and, if necessary, in real time; 2) the ability to adapt whenever changes are detected in the monitored system, allowing the application to real problems; 3) low false alarm rate and high fault isolation rate due to the robustness to outliers and noise, increasing the reliability of diagnosis. To evaluate the performance of the proposed approach in fault diagnosis, an interacting tank system simulator was used to simulate normal and several faulty conditions. Outliers and noise were added to the simulated data to evaluate the robustness of the proposed algorithms.

After this introduction, the rest of the paper proceeds as follows. Section 2 presents the theoretical concepts regarding recursive clustering algorithm, drift detection method and presents the proposed recursive clustering algorithm. Next, Section 3 presents the proposed classifier and its application in fault diagnosis. Section 4 presents the simulations and results. Finally, Section 5 presents the conclusion and suggestions for future works.

2. THEORETICAL CONCEPTS: RECURSIVE CLUSTERING ALGORITHM AND DRIFT DETECTION

2.1. Recursive Gustaffson-Kessel Algorithm

Clustering algorithms are among the most useful tools to solve pattern recognition problems, where involves analysis of nonlabeled data, or unsupervised learning (Duda, Hart, & Stork, 2001). Over the past decades, thousands of clustering algorithms have been proposed (Jain, 2010). GK algorithm, unlike many clustering algorithms that employ Euclidian distance as measure of similarity, employs Mahalanobis-like distance, which allows the identification of clusters with ellipsoidal shapes. In this algorithm the distance is defined as fol-

lows:

$$d_{ik}^2 = (x_k - v_i)A_i(x_k - v_i)^T \quad (1)$$

where d_{ik}^2 represents the distance between an input data sample $x_k = [x_{k1}, ..., x_{kn}]$, $k = 1, ..., N$, and the cluster center v_i, $i = 1, ..., c$, where N is the number of data samples, n is the number of data dimensions, and c is the number of clusters. The norm-inducing matrix $A_i, i = 1, ..., c$, defines the shape and orientation of each cluster in space. An iterative process is used in the GK algorithm to estimate the parameters of the clusters (the cluster center and fuzzy covariance matrix). This process is finished when a certain convergence criterion is reached. An extended version of the GK algorithm named evolving GK-like algorithm (eGKL) is proposed in Filev and Georgieva (2010). This approach estimates the number of clusters and performs the adaptation of its parameters recursively, maintaining the advantages of the GK algorithm. To evaluate the similarity between a new sample data and one of the existing clusters, the eGKL algorithm employs the Mahalanobis distance, defined as follows:

$$D_{ik}^2 = (x_k - v_i)F_i^{-1}(x_k - v_i)^T \quad (2)$$

where $F_i, i = 1, ..., c$ is a covariance matrix. Thus, the current data sample belongs to an existing cluster if the distance to the cluster center is smaller than the cluster radius. The eGKL algorithm uses an approach inspired in concepts of statistical process control to estimate the radius of each cluster. In this approach, it is assumed that a sample belongs to a cluster if the following relationship holds:

$$D_{ik}^2 < \chi_{n,\beta}^2 \quad (3)$$

where $\chi_{n,\beta}^2$ is the value of a Chi-squared distribution, n is the degrees of freedom and β is the confidence interval. The degrees of freedom n correspond to the input space dimension and confidence interval β is a parameter of the algorithm. This approach has the advantage of avoiding the problem called "curse of dimensionality" (Hastie, Tibshirani, & Friedman, n.d.), i.e., the problem of increasing the distance between two adjacent points with the increase in the input space dimensionality, since $\chi_{n,\beta}^2$ is proportional to the dimension of the input data. If the condition given by Eq. (3) is satisfied, it means that the current data sample belongs to a cluster, so the cluster parameters are updated. Otherwise, it is assumed that the current data sample does not belong to any one of the existing clusters, and a new cluster is created. The complete procedures of the eGLK algorithm can be seen in Filev and Georgieva (2010).

2.2. Drift Detection Method

In the literature, several drift detection methods have been proposed. In general, they can be classified into two categories: methods that perform adaptive learning at regular intervals regardless of the occurrence of changes, and methods that detect changes first and subsequently adapt the learning to these changes (Sebastião & Gama, 2009). Belonging to the second category, the DDM algorithm employs a simple method with direct application. This method is based on monitoring the number of errors produced by a learning model during prediction. The method uses the Binomial distribution to determine the general form of the probability for the random variable that represents the number of prediction errors into a sequence of n input data samples. In DDM algorithm, for each k data sample sequences, the error rate is the probability of the prediction error p_k with standard deviation $s_k = \sqrt{p_k(1 - p_k)/k}$. According to the Probability Approximately Correct (PAC) learning model (Mitchell, 1997), the error rate of the learning algorithm decreases with the increase of input data samples, and if the distribution is stationary, a significant increase in the error rate suggests context changes. In this case, it is assumed that the current model is inappropriate and should be updated. In DDM algorithm, while monitoring the error, it defines a warning and a drift level. When $p_k + s_k$ exceeds the warning level, the data samples are stored in memory. However, if $p_k + s_k$ exceeds the drift level, it is considered that there is a context change. In this situation, the model induced by the learning algorithm should be updated with the data samples stored since the time that the warning level has been reached. It is possible that the error increases and, after reaching the warning level, it decreases to lower levels. This situation corresponds to a false alarm, where there is no change of context and, therefore, no action is required and the data samples stored in the memory are no longer needed. More details about the DDM method can be found in Gama et al. (2004).

2.3. Proposed Recursive Clustering Algorithm

The algorithm proposed in this work consists of an unsupervised recursive clustering algorithm with a new mechanism of clustering update. The algorithm is a recursive version of the GK algorithm, inspired by the eGKL algorithm, and incorporating the DDM algorithm. Thus, clustering is performed in online mode and, if necessary, in real time.

Considering that there is no a priori information about the clustering structure neither a initial set of input data samples, the proposed algorithm starts by associating the center of the first cluster c_1 to the first input data sample x_1. The corresponding covariance matrix F_1, the learning rate α_1 and the number of samples associated with the first cluster M_1 are defined as follows: $c_1 = x_1$; $F_1 = F_{init}$; $\alpha_1 = \alpha_{init}$; $M_1 = 1$, where $F_{init} = \gamma I$; I is an identity matrix of n size; γ is a small positive number (default value: $\gamma = 10^-2$) and $\alpha_{init} \in [0, 1]$ is the initial learning rate (default value: $\alpha_{init} = 0.5$). If all data samples are processed, the algorithm stops, otherwise, a new input data sample x_k is obtained and the distance between the data sample and the

centers of the existing clusters is computed as:

$$D_{ik}^2 = (x_k - v_i)F_i^{-1}(x_k - v_i)^T \quad (4)$$

The similarity between the current data sample and the existing clusters is verified by the similarity condition:

$$D_{ik}^2 < \chi_{n,\beta}^2 \quad (5)$$

where $\chi_{n,\beta}^2$ is the value of a Chi-squared distribution, n is the degrees of freedom and β is the confidence interval. The degrees of freedom n correspond to the input space dimension and confidence interval β is a parameter of the algorithm. If similarity condition given by Eq. (5) is met for a cluster, it is assumed that the current sample belongs to this cluster. The cluster parameters (center, covariance matrix, learning rate and number of samples in the cluster) are then updated as follows:

$$v_q = v_q + \alpha_q(x_k - v_q) \quad (6)$$

$$F_q = F_q + \alpha_q((x_k - v_q)^T(x_k - v_q) - F_q) \quad (7)$$

$$\alpha_q = \frac{\alpha_{init}}{M_q} \quad (8)$$

$$M_q = M_q + 1 \quad (9)$$

where $q = \underset{i=1,...,c}{\arg\min}(D_{ik}^2)$. If the similarity condition given by Eq. (5) is not met, it is assumed that the current sample does not belong to any existing cluster. Then, the algorithm increments a variable that represents the number of dissimilarities, $M_{dis} = M_{dis} + 1$, then the error probability and standard deviation are computed as follows:

$$p = \frac{M_{dis}}{k} \quad (10)$$

$$s = \sqrt{p(1-p)/k} \quad (11)$$

In this algorithm, the p and s values are stored whenever $p+s$ reach the lowest value during the process, obtaining p_{min} and s_{min}. If the following condition is met:

$$p + s < p_{min} + s_{min} \quad (12)$$

then $p_{min} = p$ and $smin = s$. Note that, when algorithm starts, the p and s values must be initialized as a positive number, it is suggested set as one for each value. To decide whether the current data sample x_k represents a new cluster or it is just an outlier, warning and drift conditions are evaluated. The warning condition is verified as:

$$p + s > p_{min} + z_1 \cdot s_{min} \quad (13)$$

where z_1 is the warning level (default value: $z_1 = 2$). If the warning level is reached, then the current data sample is stored in a window of samples $W(data)_j$, $j = 1, ..., m$ (where m is the current size of the window) and then, the drift condition is evaluated. Otherwise, the algorithm processes the next input data sample. Drift condition is verified as:

$$p + s > p_{min} + z_2 \cdot s_{min} \quad (14)$$

where z_2 is the drift level (default value: $z_2 = 3$). If the drift level is reached, a new cluster is created, $c = c + 1$, and the center and the covariance matrix of the new cluster are determined by the samples stored in the data window as follows:

$$v_c = \frac{1}{m}\sum_{j=1}^{m} W(data)_j \quad (15)$$

$$F_c = cov\left(W(data)_j\right) \quad (16)$$

The remaining parameters of the new cluster (learning rate and number of samples in the cluster) are initialized as: $\alpha_c = \alpha_{init}$; $M_c = 1$.

In order to avoid redundant cluster formation, during the update, the similarity between clusters is checked. To that end, distances between the centers of the clusters are computed as follows:

$$D_{ij}^2 = (v_i - v_j)F_i^{-1}(v_i - v_j)^T \quad (17)$$

$$D_{ji}^2 = (v_j - v_i)F_j^{-1}(v_j - v_i)^T \quad (18)$$

If one of the following similarity conditions is met for two existing clusters i and j,

$$D_{ij}^2 < \chi_{n,\beta}^2 \quad (19)$$

$$D_{ji}^2 < \chi_{n,\beta}^2 \quad (20)$$

the clusters are merged. These clusters have a hyper ellipsoidal shape, defined by a mean vector, a covariance matrix, and a number of samples associated with each one. The combination of these two clusters produce a new one with parameters computed as follows (Kelly, 1994):

$$M_i = M_i + M_j \quad (21)$$

$$v_i = \frac{M_i}{M_i + M_j}v_i + \frac{M_j}{M_i + M_j}v_j \quad (22)$$

$$F_i = \frac{M_i - 1}{M_i + M_j + 1}F_i + \frac{M_j - 1}{M_i + M_j + 1}F_j + \frac{M_i M_j}{M_i + M_j(M_i + M_j - 1)}(v_i - v_j)^T(v_i - v_j) \quad (23)$$

Algorithm 1 summarizes the proposed recursive clustering algorithm.

3. PROPOSED EVOLVING FUZZY CLASSIFIER FOR FAULT DIAGNOSIS

In many current applications, the use of algorithms for pattern classification is present, such as fingerprint recognition for security systems, handwriting recognition on touch screen computers, DNA sequences identification in medical diagnostic softwares and fault diagnosis in industrial equipments. In this

Algorithm 1: Recursive Clustering Algorithm with Drift Detection

Input: $x_k,\ \chi^2_{n,\beta},\ F_{init},\ \alpha_{init},\ z_1,\ z_2$;
Output: $v_i,\ F_i$;
Read the first data sample x_1;
Initialize the first cluster;
for $k = 2, 3, ...$ **do**
  Read x_k;
  Compute D^2_{ik} for all clusters;
  Identify the closest cluster;
  if $D^2_{qk} < \chi^2_{n,\beta}$ **then**
    Update the closest cluster;
  else
    Update the dissimilarity number M_{dis};
    Compute p and s;
    if $p + s < p_{min} + s_{min}$ **then**
      Update p_{min} and s_{min};
    end if
    if $p + s > p_{min} + z_1 \cdot s_{min}$ **then**
      Store x_k in the data window $W(data)_j$;
    end if
    if $p + s > p_{min} + z_2 \cdot s_{min}$ **then**
      Create new cluster;
    end if
  end if
  Compute D^2_{ij} and D^2_{ji} for all clusters;
  if $D^2_{ij} < \chi^2_{n,\beta}$ *or* $D^2_{ji} < \chi^2_{n,\beta}$ **then**
    Merge redundant clusters;
  end if
end for

context, the problem of pattern classification consists in assigning a class or a category for each data sample from a set of "raw" data (Duda et al., 2001). Pattern classification algorithms based on fuzzy rules have been used in many applications due to their advantages in relation to classic algorithms for pattern classification, especially by the good prediction performance in real problems and good transparency in linguistic rules (Jang, Sun, & Mizutani, 1997), which allows an easy comprehension of the dependence between pattern characteristics. The typical architecture of a fuzzy classifier consists of a set of IF ... THEN fuzzy rules, defined as:

$$\text{RULE}_i:\ \text{IF}\ x_1\ \text{IS}\ \mu_{i1}\ \text{AND}\ ...\ \text{AND}\ x_n\ \text{IS}\ \mu_{in}\ \text{THEN}\ y_i = L_i \tag{24}$$

where $[x_{k1}, ..., x_{kn}]$ are the input variables or patterns of n dimensionality; $[\mu_{i1}, ..., \mu_{in}]$ are antecedent fuzzy sets of the *i*th fuzzy rule; y_i is the output; L_i is the crisp output corresponding to the class label from the set $[1, ..., K]$, where K is the number of classes. For each new input data sample x_k, the classification is obtained by assigning to it the label of the class associated with the rule having the highest activation degree. The class is determined as follows:

$$y_i = L_{i^*} \tag{25}$$

where $i^* = \arg\max_{1<i<R}(\tau_i)$; R is the number of fuzzy rules and τ_i is the activation degree of the *i*th fuzzy rule, defined by a t-norm, usually expressed as a product operator:

$$\tau_i = \prod_{j=1}^{n} \mu_{ij}(x_j) \tag{26}$$

where μ_{ij} are the membership functions of fuzzy sets defined by Gaussians:

$$\mu_{ij} = e^{-\frac{1}{2}\left(\frac{(x_j - v_{ij})^2}{\sigma^2_{ij}}\right)} \tag{27}$$

where v_{ij} and σ^2_{ij} represent respectively the membership functions center and variance. Usually, to implement this fuzzy classifier architecture, clustering is performed in the input and/or output space. Then, rules are created using one-dimensional (or univariate) fuzzy sets, generated from the projection of the clusters in the axis of each variable. According to Lemos et al. (2011), this approach can lead to information loss if there is interaction between variables, and to avoid this, the authors propose the use of multivariate Gaussian membership functions to represent antecedent fuzzy sets of each rule. These membership functions are described as:

$$H(x) = e^{-\frac{1}{2}\left((x-v)\Sigma^{-1}(x-v)^T\right)} \tag{28}$$

where v is a $1 \times n$ central vector and Σ is a $n \times n$ symmetric positive definite matrix. The central vector is defined as the modal value and represents $H(x)$ typical value and the Σ matrix denotes the dispersion and represents $H(x)$ spreading. Thus, each cluster found by the clustering algorithm is associated with a fuzzy rule and the multivariate Gaussian membership function parameters are defined as the parameters of the corresponding cluster. If multivariate Gaussian membership functions are used, the fuzzy classifier will have a rule set defined as:

$$\text{RULE}_i:\ \text{IF}\ x_k\ \text{IS}\ A_i\ \text{THEN}\ y_i = L_i \tag{29}$$

where A_i is the fuzzy set with multivariate Gaussian membership function of the *i*th fuzzy rule, with parameters extracted from the corresponding cluster. In general, more than one rule can be used to describe a class, e.g, the class can be multimodal. In this case, only one rule cannot be sufficient to describe all possible variations of the same class. Thus, the fuzzy classifier aggregates rules outputs associated with the same class using a s-norm. The result of the aggregation can be interpreted like rules as follows:

$$(\text{IF}\,x_k\,\text{IS}\,A_i)\text{OR}(\text{IF}\,x_k\,\text{IS}\,A_j)\text{OR}...(\text{IF}\,x_k\,\text{IS}\,A_k)\text{THEN}\,y_i = L_i \tag{30}$$

This aggregation results in the degree of relevance of each known class. The classification of each new sample x_k is defined by the class with the highest relevance degree.

Data samples classes are not known a *priori* in some pattern classification applications. In these situations it is required

the use of an unsupervised learning process for classifier implementation. Moreover, in applications where the pattern classification should be performed in real time, the learning should be performed using incremental algorithms, processing each data sample once as a data stream. To solve these problems, the solution is to use a recursive clustering algorithm. We propose in this paper an evolving fuzzy classifier based on recursive clustering algorithm with drift detection presented in Section 2.3, which allows the creation of a fuzzy rule base in online mode and, if necessary, in real time from input data samples. This approach is different from the ones employed in traditional fuzzy classifiers, which require some training (usually supervised) conducted in off-line mode. For rule base update, the proposed evolving fuzzy classifier uses the output of the recursive clustering algorithm described in the previous section. For each new input data sample, if a new cluster is created, a new fuzzy rule given by Eq. (29) is added to the rule base, where the cluster parameters are used as parameters of the multivariable Gaussian membership function of the antecedents. The rule consequent (the crisp output corresponding to the class label) must be defined by experts or system operators, since in unsupervised learning processes incoming online samples usually are not pre-labelled. If a cluster is updated, the corresponding class label is determined as the consequent of the fuzzy rule with the highest activation degree, and the user intervention is not necessary. If two clusters are merged by the recursive clustering algorithm, the corresponding fuzzy rules are also merged to represent an unique class. It should be noted that, both the number of rules and the number of classes are determined during the evolving process, and it is not necessary to set these parameters a *priori*. Algorithm 2 summarizes the procedures of the classifier.

Algorithm 2: Evolving Fuzzy Classifier

Input: x_k;
Output: y_k;
Initialize the classifier;
for $k = 1, 2, ...$ **do**
  Read x_k;
  Execute the recursive clustering algorithm with drift detection;
  if *new cluster is created* **then**
    Create new fuzzy rule;
    Define the new class elicited by expert / system operator;
    y_k = label of the new class;
  end if
  if *cluster is updated* **then**
    Update the corresponding fuzzy rule;
    Find the most active rule;
    y_k = label of the most active rule;
  end if
  if *clusters are merged* **then**
    Merge the corresponding fuzzy rules;
  end if
end for

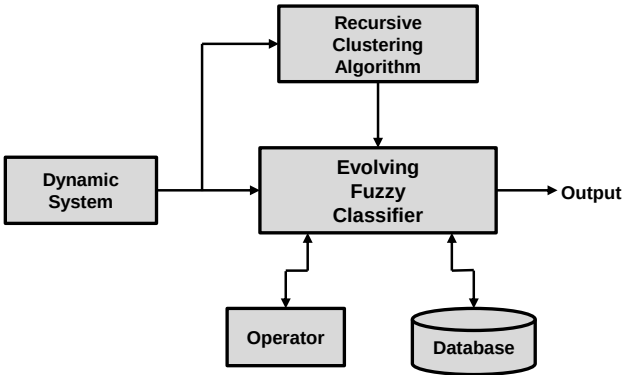


Figure 1. Fault diagnosis with the evolving fuzzy classifier.

Figure 1 illustrates the application of the proposed classifier for fault diagnosis. Data samples are obtained from a dynamic system in a continuous stream, usually provided by sensors that monitor the process. These data might require the use of pre-processing techniques for feature extraction. The rule set of the classifier starts empty at the beginning. Rules are created as the recursive clustering algorithm creates clusters to represent the data stream. Each rule will be related to a class, and each class will be related to a dynamic system condition, representing a normal operation or a fault. When a new rule is created, the system operator is notified and informs the label of the new class that defines it as a normal operation condition or as a specific fault. All of the necessary diagnostic information, the fuzzy rules and classes label, are stored in a unified database and updated while the system is used. The classifier database will contain a set of fuzzy rules and classes labels defined after an initial period of operation. When a new data sample is associated with an existing cluster, the classifier updates the corresponding fuzzy rule and classifies the dynamic system condition as the label present in the consequent of the fuzzy rule with the highest activation degree. It should be noted that, in this situation, user intervention is not required, and the classification of the dynamic system condition is performed automatically. The main feature of the classifier proposed in this work is ability to diagnose faults in a complex non-stationary dynamic system in online mode and, if necessary, in real time. The classifier does not require any a *priori* information about the dynamic model neither process historical data. This allows the classifier to construct a rule base in an evolving way and, with the aid of the operator, to learn to diagnose faults as they occur. Thus, the proposed classifier is able to adapt to the dynamic system, making it possible to diagnose faults not previously known.

4. SIMULATIONS AND RESULTS

The proposed classifier was evaluated for fault diagnosis in an interacting tank system. The interacting tank system model

employed in this work was based in the system proposed by Braga, Jota, Polito, and Pena (1995) and allows to simulate faults that resembles the faults of real industrial plants. As illustrated in Fig. 2, the system comprises of a reservoir (TQ-1) and two passively interconnected tanks (TQ-2 and TQ-3). Using the interacting tank system model is possible to perform fault simulation on the actuators (pneumatic valves and pumps), at the system components (connection pipes between tanks) and on the sensors, with different sets of parameters. The types of faults are detailed in Table 1. In the fault simulation, the system starts at normal operation, and a fault is set at half of the simulation interval. Figure 3 shows as an example the curves of the TQ-2 level, TQ-3 level, TQ-2 input flow rate and TQ-3 output flow rate in fault simulation (FCV-1 valve tightness). At the beginning of each simulation, the system is working under normal operation, and the fault starts at the half of the period.

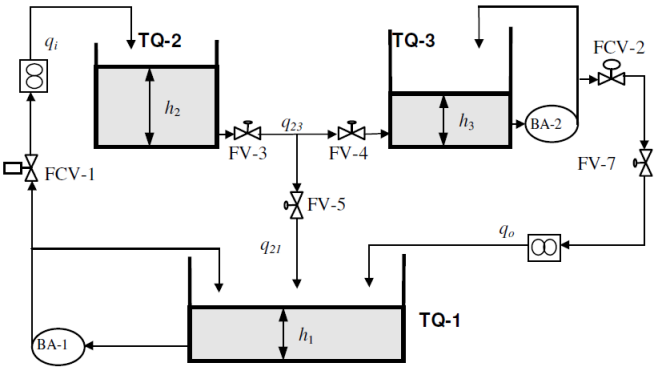


Figure 2. Representation of the interacting tank system.

Table 1. Types of faults on interacting tank system.

Index	Description
0	Normal operation
1	FCV-1 valve tightness
2	FCV-2 valve tightness
3	BA-1 pump shutdown
4	BA-2 pump shutdown
5	pipe clogging between TQ-1 and TQ-2
6	pipe clogging between TQ-1 and TQ-3
7	pipe clogging between TQ-2 and TQ-3
8	pipe leakage between TQ-2 and TQ-3
9	TQ-3 level sensor fault
10	TQ-3 output flow rate sensor fault
11	TQ-2 input flow rate sensor fault

Different scenarios were used in the fault diagnosis experiments. Each scenario consists in the simulation of sequences from 3 to 11 randomly selected fault types within a set of

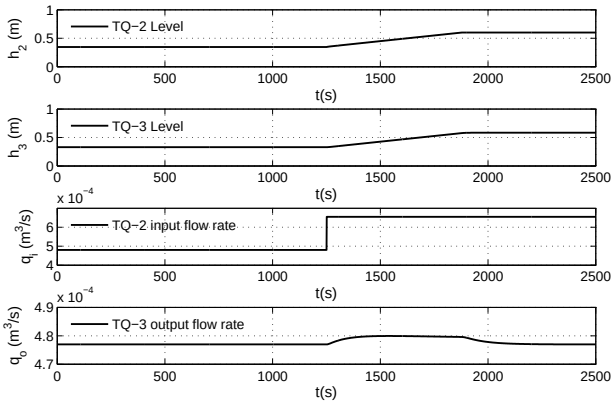


Figure 3. Fault Simulation: FCV-1 valve tightness.

faults with periods of normal operation between faults. In order to assess the robustness of the proposed classifier to the presence of noise in the data, for each monitored variable random Gaussian noise was added with a zero mean and standard deviation equal to 1% of the variable nominal value, considering normal operation of the system. As inputs of the classifier were provided in an online mode data samples related to monitored variables of the interacting tank system: TQ-2 level, TQ-3 level, TQ-2 input flow rate and TQ-3 output flow rate. For each fault sequence, the output classifier was compared to the sequence provided. Whereas the classifier starts with no fuzzy rule set, the first samples of data should match the normal operation of the system, i.e., the first rule created to describe the normal operation. For the experiments, the parameters of the recursive clustering algorithm were defined as: $\chi^2_{n,\beta} = 9.4877$; $F_{init} = 10^{-2}I$; $\alpha_{init} = 0.5$; $z_1 = 2$; $z_2 = 3$.

Figure 4 show as an example the results of fault diagnosis in 5 faults scenario simulated scenario, where we can compare the estimated output (classified faults sequence) of the proposed classifier with the desired output (selected faults sequence) from input data samples. Results show that the classifier was able to correctly diagnose all the interacting tank system faults. Whereas the presence of noise in the data samples, the occurrence of false alarms or misclassification (represented by isolated points on the graph) is low, even in the scenario with the highest number of possible faults.

The classifier performance evaluation in this work was held in terms of faults detection and fault classification, as suggested in Vachtsevanos et al. (2006). Three metrics were calculated in fault detection evaluation: Probability of Detection (POD), Probability of False Alarm (POFA) and Accuracy (ACC). Regarding fault classification evaluation, the metric Fault Isolation Rate (FIR) was used. Other metrics that were used to assess the performance of the proposed classifier are: Detection

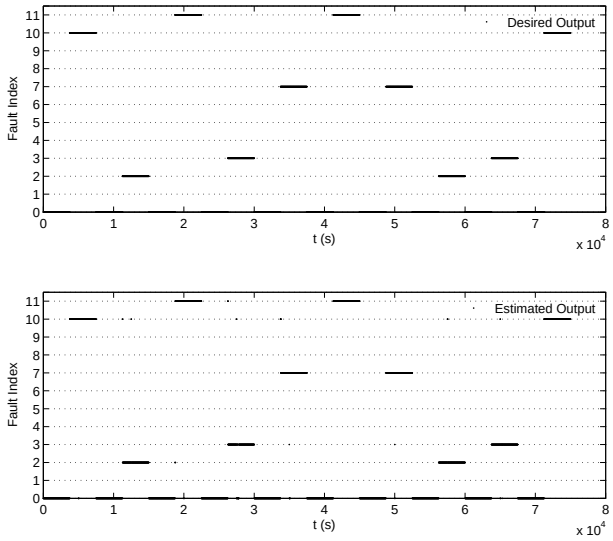


Figure 4. Desired output and estimated output by proposed classifier in 5 faults scenario.

Delay Time (DDT), Isolation Delay Time (IDT) and Operator Intervention Rate (OIR). All results of fault diagnosis experiments with interacting tank system obtained by classifier proposed in this work were compared to the results obtained using the evolving fuzzy classifier proposed by Lemos et al. (2013). For the experiments, the parameters of this alternative classifier were set to: $w = 100$, $\lambda = 0.001$, $\alpha = 0.01$, $T_{\mu_y} = 0.01$. According to authors, this combination has been found experimentally to provide a good balance between the false alarm rate and the sensibility of the fault detection and diagnostic approach.

Table 2 summarizes the results for both classifiers using the fault detection metrics described. The results show that the classifier proposed in this work has higher levels of fault detection rates and accuracy in all scenarios, and no occurrence of false alarm. These results prove the efficiency of the algorithm in detecting simulated faults in the interacting tank system. Despite its lower fault detection rates and lower accuracy, the classifier proposed by Lemos et al. (2013) also not showed any false alarms.

Table 3 summarizes the results for both classifiers using the faults classification metrics described. The results show that the classifier proposed in this work presented higher fault isolation rate in all scenarios. In all scenarios the operator intervention on faults classification was very low. These results shows the ability of the classifier to automatically diagnose almost all faults after the first occurrence, and it also reveals their ability to learn. Note that, in general, the classifier proposed by (Lemos et al., 2013) had a lower performance in faults classification than the proposed classifier and it needed more operator interventions.

Table 4 summarizes the results for both classifiers using the time metrics in fault detection and classification. A comparison between the average values for fault detection time and fault isolation time demonstrates that faults classification is faster after the first occurrence of each type of fault, since the classifier database already has the fuzzy rules and labels for all types of detected faults, not requiring an operator intervention. The results of the experiments with the classifier proposed by Lemos et al. (2013) demonstrated a faster response than the classifier proposed in this work, which is related to different update mechanisms in the clustering algorithms used in each one of the classifiers.

Table 2. Faults detection performance.

Scenario	Proposed		
	POD (%)	POFA (%)	ACC (%)
3 faults	99.38	0.00	99.67
5 faults	99.25	0.00	99.63
7 faults	99.53	0.00	99.67
9 faults	99.12	0.00	99.56
11 faults	99.20	0.00	99.60
Scenario	Lemos et al. (2013)		
	POD (%)	POFA (%)	ACC (%)
3 faults	89.35	0.00	94.67
5 faults	83.04	0.00	91.75
7 faults	82.27	0.00	91.10
9 faults	79.78	0.00	89.89
11 faults	76.02	0.00	88.01

Table 3. Faults classification performance.

Scenario	Proposed		Lemos et al. (2013)	
	FIR (%)	OIR (%)	FIR (%)	OIR (%)
3 faults	99.55	0.05	94.67	0.28
5 faults	96.76	0.04	91.88	0.29
7 faults	94.24	0.03	90.30	0.30
9 faults	92.69	0.03	89.86	0.31
11 faults	91.43	0.03	88.01	0.31

Table 4. Fault detection and classification time.

Scenario	Proposed		Lemos et al. (2013)	
	DDT (s)	IDT (s)	DDT (s)	IDT (s)
3 faults	0.065	0.003	0.015	0.003
5 faults	0.753	0.680	0.017	0.003
7 faults	1.482	1.321	0.021	0.004
9 faults	1.936	1.826	0.018	0.004
11 faults	2.327	2.204	0.018	0.004

To evaluate the robustness of the proposed classifier in the presence of outliers in the data, another experiment was conducted. In this experiment, a 5 faults scenario was simulated. Outliers were inserted in the data samples, i.e., some samples were corrupted with high variance noise. Even in the

presence of outliers, the fault diagnosis results for this experiment shows that the proposed classifier was able to correctly detect and diagnose all faults considered. This result shows that the classifier was able to correctly distinguish between outliers and valid data samples. The results of this experiment are presented in Table 5 and Table 6. Analysing these tables, one can note that the proposed classifier has virtually the same performance in fault diagnosis with absence or presence of outliers, and also not showed occurrence of false alarm. This experiment showed the greater robustness of the classifier proposed in this work when compared with the classifier proposed by Lemos et al. (2013), since the latter showed major differences in fault detection and fault classification rates in scenarios with and without outliers.

Table 5. Faults detection performance with outliers.

Scenario	Proposed		
	POD (%)	POFA (%)	ACC (%)
without outliers	99.25	0.00	99.63
with outliers	99.26	0.00	99.63
Scenario	Lemos et al. (2013)		
	POD (%)	POFA (%)	ACC (%)
without outliers	83.78	0.00	91.75
with outliers	79.00	0.00	89.51

Table 6. Fault classification performance with outliers.

Scenario	Proposed		Lemos et al. (2013)	
	FIR (%)	OIR (%)	FIR (%)	OIR (%)
without outliers	96.73	0.04	91.88	0.30
with outliers	96.34	0.04	89.00	0.32

5. CONCLUSION

An evolving fuzzy classifier for fault diagnosis of dynamic systems was presented in this work. The proposed classifier is composed by a set of fuzzy rules created and updated based on recursive clustering algorithm. A new mechanism for cluster updating based on a drift detection method is employed, where the update of the cluster depends not only of the similarity measure, but also on the data context monitoring. As suggested by the simulation results, this feature gives the proposed classifier robustness to outliers and noise. An interacting tank system model was used for evaluation of the classifier proposed in this work. The classifier was able to detect and classify all faults with a high performance, even in the presence of outliers and noise. The high fault isolation rate and low false alarm rate obtained in all simulated scenarios showed that the recursive clustering algorithm with drift detection method was able to efficiently distinguish data samples representing clusters of invalid data. Moreover, the proposed classifier was able to automatically diagnose almost all faults, requiring operator intervention on a small percentage of cases. This demonstrates the advantage of the continuous and incremental learning of the classifier over other classifiers that require retraining whenever an unknown type of fault is found. The classifier proposed in this work has as advantages: the ability to learn from faults in online mode and in real time; the ability to adapt to cope with changes in the dynamic system; and robustness to the presence of outliers and noise in the input data. Summarizing, the proposed classifier has showed to be a promising alternative for application in fault diagnosis where other methods prove to be inefficient or less advantageous, because of the characteristics of such systems. In a future work, we will investigate the application of the proposed algorithm in the real time fault diagnosis and prognosis of industrial machines and equipments.

REFERENCES

Abellan-Nebot, J. V., & Subirón, F. R. (2010). A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, *47*, 237-257. doi: 10.1007/s00170-009-2191-8

Angelov, P., & Filev, D. (2003). On-line Design of Takagi-Sugeno Models. In T. Bilgiç, B. D. Baets, & O. Kaynak (Eds.), *Fuzzy Sets and Systems — IFSA 2003* (Vol. 2715, p. 576-584). Springer Berlin Heidelberg. doi: 10.1007/3-540-44967-1-69

Angelov, P., Filev, D., & Kasabov, N. (2010). *Evolving Intelligent Systems: Methodology and Applications*. New York, USA: John Willey & Sons.

Braga, A. R., Jota, F. G., Polito, C. M., & Pena, R. T. (1995). Development of an interacting tank system for the study of advanced process control strategies. In *Proceedings of the 38th Midwest Symposium on Circuits and Systems* (Vol. 1, p. 441-444). doi: 10.1109/MWSCAS.1995.504471

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. New York,USA: John Wiley & Sons.

Filev, D., Chinnam, R. B., Tseng, F., & Baruah, P. (2010). An Industrial Strength Novelty Detection Framework for Autonomous Equipment Monitoring and Diagnostics. *IEEE Transactions on Industrial Informatics*, *6*(4), 767-779. doi: 10.1109/TII.2010.2060732

Filev, D., & Georgieva, O. (2010). An extended version of the Gustafson–Kessel algorithm for evolving data stream clustering. In P. Angelov, D. Filev, & N. Kasabov (Eds.), *Evolving Intelligent Systems: Methodology and Applications* (p. 273-300). New York, USA: John Wiley & Sons.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A. L. C. Bazzan & S. Labidi (Eds.), *Advances in Artificial Intelligence – SBIA 2004* (p. 286-295). Springer Berlin Heidelberg. doi: 10.1007/978-3-540-28645-5-29

Gustafson, D. E., & Kessel, W. C. (1979). Fuzzy clustering with fuzzy covariance. In *Proceedings of IEEE Conference on Decision and Control* (p. 761-766).

Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York, USA: Springer-Verlag.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. doi: 10.1016/j.patrec.2009.09.011

Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*. Upper Saddle River, USA: Prentice-Hall, Inc.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. doi: 10.1016/j.ymssp.2005.09.012

Kasabov, N., & Song, Q. (2002). DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time Series Prediction. *IEEE Transactions on Fuzzy Systems*, *10*(2), 144-154. doi: 10.1109/91.995117

Kelly, P. M. (1994). *An algorithm for merging hyperellipsoidal clusters* (Tech. Rep.). LA-UR-94-3306, Los Alamos National Laboratory, Los Alamos, NM.

Lemos, A., Caminhas, W., & Gomide, F. (2011). Multivariable Gaussian Evolving Fuzzy Modeling System. *IEEE Transactions on Fuzzy Systems*, *19*(1), 91-104. doi: 10.1109/TFUZZ.2010.2087381

Lemos, A., Caminhas, W., & Gomide, F. (2013). Adaptive Fault Detection and Diagnosis Using an Evolving Fuzzy Classifier. *Information Sciences*, *220*, 64-85. doi: DOI: 10.1016/j.ins.2011.08.030

Leng, G., McGinnity, T. M., & Prasad, G. (2005). An Approach for On-Line Extraction of Fuzzy Rules Using a Self-Organising Fuzzy Neural Network. *Fuzzy Sets & Systems*, *150*(2), 211–243. doi: DOI: 10.1016/j.fss.2004.03.001

Lima, E., Hell, M., Gomide, F., & Ballini, R. (2010). Evolving fuzzy modeling using participatory learning. In P. Angelov, D. Filev, & N. Kasabov (Eds.), *Evolving Intelligent Systems: Methodology and Applications* (p. 67-87). New York, USA: John Wiley & Sons.

Lughofer, E. (2008). FLEXFIS: A robust Incremental Learning Approach for Evolving Takagi–Sugeno Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, *16*(6), 1393-1410. doi: DOI: 10.1016/j.fss.2004.03.001

Lughofer, E., & Guardiola, C. (2008). Applying Evolving Fuzzy Models with Adaptive Local Error Bars to On-line Fault Detection. In *Proceedings of 3rd International Workshop on Genetic and Evolving Fuzzy Systems - GEFS 2008* (p. 35-40). doi: 10.1109/GEFS.2008.4484564

Mitchell, T. M. (1997). *Machine Learning*. New York, USA: McGraw-Hill.

Rong, H. J., Sundararajan, N., Huang, G. B., & Saratchandran, P. (2006). Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets & Systems*, *157*(9), 1260-1275. doi: 10.1016/j.fss.2005.12.011

Sebastião, R., & Gama, J. (2009). A Study on Change Detection Methods. In *Proceedings of 4th Portuguese Conference on Artificial Intelligence* (p. 353-364).

Soleimani-B., H., Lucas, C., & Araabi, B. N. (2010). Recursive Gath-Geva clustering as a basis for evolving neuro-fuzzy modeling. *Evolving Systems*, *1*(1), 59-71. doi: 10.1007/s12530-010-9006-x

Vachtsevanos, G., Lewis, F., Roeme, M., Hess, A., & Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, USA: John Wiley & Sons.

Venkatasubramanian, V. (2005). Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities. *Computers and Chemical Engineering*, *29*(6), 1253-1263. doi: 10.1016/j.compchemeng.2005.02.026

BIOGRAPHIES

Maurilio Inacio received the B.Sc. degree in computing engineering from the Faculdade de Ciencia e Tecnologia de Montes Claros, Montes Claros, Brazil, in 2007, and the M.Sc. degree in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2010, where he is currently working toward the Ph.D. degree with the Department of Electrical Engineering. His current research interests include evolving intelligent systems, fault diagnosis and prognosis.

Andre Lemos received the B.Sc. degree in computer science from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2003, the M.Sc. degree and the Ph.D. degree in electrical engineering from the same university in 2007 and 2011, respectively. He is currently an Associate Professor with the Department of Electronic Engineering, Federal University of Minas Gerais. His current research interests include adaptive and evolving intelligent systems, machine learning and applications in time-series forecasting, fault detection and diagnosis, and nonlinear systems modeling.

Walmir Caminhas received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 1987 and 1989, respectively, and the Ph.D. degree from the University of Campinas, Campinas, Brazil, in 1997, all in electrical engineering. He is currently an Associate Professor with the Department of Electronic Engineering, Federal University of Minas Gerais. His research interests include computational intelligence and fault detection in dynamic systems.

Novel Real-Time Nondestructive Technology for Chemical and Structural Health Management of Solid Rocket Propellants

Sami Daoud[1], Michal J. Villeburn[2], Kevin D. Bailey[3], Gordon Kinloch[4]

[1,2,3] *Raytheon Missile Systems, Tucson, Arizona, 85734, United States of America*
sami_daoud@raytheon.com
mjvilleburn@raytheon.com
kdbailey@raytheon.com

[4]*United Kingdom Ministry of Defense, Bristol, BS34 8JH, United Kingdom*

DESBVRAAMPIM2a@mod.uk

ABSTRACT

An innovative prognostics and chemical health management (CHM) technique was developed, for quantifying and characterizing health status of a CL-01 composite solid rocket propellant of tactical rocket motors. The technique is a cutting-edge real-time nondestructive technology approach which utilizes Near Infrared (NIR) spectra (M. Blanco, and I. Villarroya, 2002) emitted by microPHAZIR™ NIR miniature handheld platform, developed by Thermo Fisher Scientific. Benchtop high-performance liquid chromatography (HPLC) and ion chromatography (IC) were utilized as baseline reference techniques for correlation to microPHAZIR™ NIR measurements.

To build a quantitative calibration model, near infrared spectra were acquired for twenty freshly manufactured mixes of CL-01 propellant formulae, which were iterated using a D-Optimal full-factorial design of experiment (DOE). Four-hundred eighty measurements were recorded and analyzed using Partial Least Squares (PLS) regression analysis for model building and method development (Schreyer, 2012). NIR results were correlated to spectra, which were produced using HPLC and IC reference techniques and were determined to be in precise agreement. All recorded measurements that were performed using microPHAZIR™ handheld platform were successfully validated with HPLC and IC measurements. An algorithm was developed for microPHAZIR™ NIR thus qualifying the platform as a real-time nondestructive test (NDT)/ nondestructive evaluation (NDE) tool for quantification of primary chemical constituents of CL-01 composite solid rocket propellant. Primary chemical constituents of CL-01 comprise a binder, oxidizer, plasticiser, and antioxidant/stabilizer.

Data sets for Shore-A hardness of each of the twenty DOE mixes were collected and used to calculate elastic modulus, tensile strength and percent strain. Calculated results conformed to specification requirements for CL-01 solid rocket propellant, henceforth confirming use of Shore A hardness as a real-time nondestructive test technique for validation of structural health of a solid rocket propellant.

This teaming effort between Raytheon Missile Systems (RMS), United Kingdom Ministry of Defence (UK MoD), Alliant Techsystems Launch systems (ATK LS), and Thermo Fisher Scientific demonstrated outstanding ability to utilize miniature cutting-edge technology to perform real-time NDT of CL-01 composite solid rocket propellant without generating chemical waste and residue and to ameliorate RMS technology base to capture incipient failures before the fact. The new technique will further be adapted for use to measure primary chemical constituents of other solid rocket propellants, liquid propellants, and composite explosives. The new technique will significantly reduce costs associated with surveillance and service life extension programs (SLEPs), which are often destructive and requires use of lengthy and expensive test techniques described in North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG)-4170 and Allied Ordnance Publication (AOP)-7 manuals.

1. INTRODUCTION

Tactical missiles are often exposed to severe thermal and dynamic stressors, often associated with long-term exposure to harsh environments, during transportation handling, transportation vibration, ejection and launch shock, diurnal cycling, storage, or when fielded. These stressors may act

Sami Daoud et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

individually or synergistically to factor into the aging, deterioration, and eventual decommissioning of critical warfighting assets. As a result, adverse impact and henceforth degradation reliability and/or safety of the assets may occur and will affect the total life cycle cost of fielding these weapon systems in a high state of readiness. Reliability analyses of legacy data indicated failure occurrences in missile structures, energetic and electronic components, all often associated with long-term exposure to static (heat, humidity, salt, etc.), and dynamic (transportation shocks, vibration, etc.) stressors.

Today's most common methods of NDT for evaluating the health of energetic systems are radiographic (X-ray imaging, X-ray computed tomography (CT), etc.), electrical (Eddy-current and electro-magnetic methods), dye penetrant, acoustic and ultrasonic, or a combination thereof. These methods are used by manufacturers during production processes, and mostly for quality control. Moreover, for fielded tactical missile systems, these methods are impractical for use. For health monitoring in the field, deployable portable platforms such as microPHAZIR™ NIR handheld platform become valuable as NDT/NDE tools.

A joint teaming effort was carried out between the UK Ministry of Defence (MoD), Raytheon Missile Systems, ATK Launch Systems, and Thermo Fisher Scientific to qualify microPHAZIR™ NIR platform as a miniature portable real-time NDE tool. The effort was successfully executed and would enable RMS, other defense contractors, US DoD and UK MoD to quantify primary chemical constituents of CL-01 solid rocket propellant nondestructively and on real-time basis. CL-01 is a composite high volumetric ballistic potential solid rocket propellant used in the propulsion subsystem of tactical missiles. Successful qualification of microPHAZIR™ NIR platform to quantify primary chemical constituents of CL-01 will enable defense contractors, DoD and MoD personnel to adapt this technology to quantify chemical constituents of all composite solid rocket propellants, liquid propellants, and warhead explosives, henceforth institute a cutting-edge technology of chemical health management (CHM).

Concurrently, RMS under the direction of UK MoD has successfully validated a new technique for determining structural health of CL-01 solid rocket propellant, also nondestructively and on real-time basis, henceforth integrating structural health management (SHM) with chemical health management (CHM). The combined techniques introduce a novel approach to prognostics and health management (PHM) of composite solid rocket propellants and warhead explosives.

The proposed technology is a proactive real-time NDE/NDT technique which replaces the old destructive test methodologies, described in NATO STANAG-4170 and AOP-7 manuals, imposed by Surveillance and Life Extension Programs (SLEPs) of past and present day techniques. The proposed technology is a novel cutting-edge achievement as a NDT tool, in that it will define new means for quantifying chemical constituents of multi-component solid rocket propellant formulae while at the same time shedding light on propellants structural health, and will enable the manufacturer to define the anticipated residual useful life (RUL) of solid rocket propellants from a chemical as well as structural perspective. This achievement will shed valuable information about the anticipated mechanical and structural behavior of the solid rocket propellant matrix, in what is often referred to as "the chemical-mechanical link". The combination of chemical and mechanical (structural) health of the solid rocket propellant is the definition of prognostics and health management (PHM) and is the basic principle which will define whether a rocket motor (propulsion subsystem) would be warranted as "safe and suitable for service (S3)".

Today RMS and the UK MoD surveillance strategies seek to extend time between periodic surveillances, henceforth reducing tasks associated with subsystem breakdown, test and criticality analysis (BTCA) by as much as 50% or more. On average, a surveillance program is often recommended once every 3 to 4 years on a sample population which represents the fielded and/or stored weapons inventory. With the introduction of microPHAZIR™ NIR real-time technology and associated structural health management, it will be feasible to extend the time between surveillance programs activities and/or reduce the number of assets that have to undergo surveillance. When a SLEP plan is established for solid rocket motor (propulsion subsystem) subsystem, complex steps must be executed and comprise disassembly, dissection and extraction of propellant samples, followed by extensive testing (physical, chemical, hazards, and mechanical tests) of the rocket motor solid rocket propellant matrix, often referred to as "breakdown, test and criticality analysis (BTCA)". BTCA coupled with arena testing (static fire) of the solid rocket motor as well as other subsystems are challenging tasks, from manpower, cost and schedule perspectives, and therefore the need to exercise cost controls while at the same time maintain absolute confidence in assets health demand that novel technology approaches such as those associated with microPHAZIR™ NIR platform and more advanced (exploratory) technologies become integral part of SLEP. The ultimate goal is to be able to (i) predict subsystems, and henceforth system anomalies proactively and sufficiently in advance to institute corrective actions and/or preventive measures; (ii) ensure that the subsystem is reliable from a performance as well as safety perspective to fulfill warfighters requirements; and (iii) reduce generated chemical waste, logistics footprint, logistics response time, and life-cycle costs, which will ultimately increase systems availability, and enhance customer-supplier business relationship.

The ultimate goal of the proposed technology will be that of enabling RMS and the customer (UK MoD) to realize early warnings of unsafe conditions using real-time data, collected via microPHAZIR™ NIR miniature handheld platform and other advanced technologies of Thermo Fisher Scientific. Gaining real-time knowledge about the current health of a propellant or explosive matrix will offer effective insight to predicting residual useful life (RUL) of the system and its corresponding inventory.

Successful application of microPHAZIR™ NIR handheld platform as a NDE/NDT tool is the cornerstone and the spring board for future development of PHM of energetic subsystems: Cartridge Actuated Devices (CADs), Propellant-Actuated Devices (PADs), and electro-explosive devices (EEDs) of tactical and strategic missiles. MicroPHAZIR™ NIR handheld platform offers enormous potential for applications requiring real-time monitoring of the health status of warheads and solid rocket motors which have been exposed to fatigue resulting in chemical and mechanical (structural) degradation.

2. EXPERIMENTAL

A D-Optimal Design of Experiment (DOE) was initiated using Minitab 16, with the goal of manufacturing twenty laboratory scale mix iterations of CL-01 (L. Biegert, and B. Cragun, 2013) solid rocket propellant. The twenty mix iterations are listed in Table 1.

Table 1: DOE Design Series for Primary Constituents

Mix Number RBC1691-99-	Mix #	Oxidizer	Plasticizer	Antioxidant	Binder	TOTAL
38 (Bay 8)	1	84	3.25	0.350	12.400	100
39 (Bay 5)	2	84	3.00	0.275	12.725	100
40 (Bay 8)	3	82	2.75	0.300	14.950	100
41 (Bay 5)	4	85	2.50	0.300	12.200	100
42 (Bay 8)	5	83	2.50	0.325	14.175	100
43 (Bay 5)	6	85	3.25	0.275	11.475	100
44 (Bay 8)	7	83	3.00	0.300	13.700	100
45 (Bay 5)	8	81	3.50	0.350	15.150	100
46 (Bay 8)	9	82	3.00	0.350	14.650	100
47 (Bay 5)	10	82	3.50	0.275	14.225	100
48 (Bay 8)	11	83	2.75	0.350	13.900	100
49 (Bay 5)	12	83	3.50	0.250	13.250	100
50 (Bay 8)	13	85	3.00	0.250	11.750	100
51 (Bay 5)	14	81	2.50	0.275	16.225	100
52 (Bay 8)	15	85	2.75	0.325	11.925	100
53 (Bay 5)	16	81	2.75	0.250	16.000	100
54 (Bay 8)	17	81	3.25	0.325	15.425	100
55 (Bay 5)	18	84	2.50	0.250	13.250	100
56 (Bay 8)	19	82	3.25	0.250	14.500	100
57 Bay 5)	20	84	3.50	0.300	12.200	100

For each mix, constituents were varied above and below specification limits (+1, -1), to capture acceptable high and low limits of each constituent within the formulation.

All CL-01 propellant raw materials were procured from ATK Allegany Ballistics Laboratory (ATK ABL) and were certified to material specifications. Raw materials which were utilized in the manufacture of the twenty DOE mixes are listed in Table 2. No hazards tests were required since all manufactured formulations were within the history of material sets prepared at ATK Launch Systems.

Table 2: CL-01 Propellant Raw Material Specifications

Material/Chemical	Specification
Prepolymer	MIL-H-85497
Oxidizer	MIL-A-82667
Curative	MIL-C-85498, Type II
Bonding Agent 1	HS 6-0823
Bonding Agent 2	MY 0510 HS 6-0093
Plasticizer	S273020C
Antioxidant	HS 6-0091
Combustion Stabilizer	MIL-Z-85500
Opacifier	HS 6-0090

Each of the twenty mix iterations amounted to 600 grams (1-pint each). Each of the twenty CL-01 propellant mixes was prepared using a 1-pint Baker Perkins mixer. At the conclusion of the mix cycle, each of the twenty 1-pint mixes was vacuum-cast into a Teflon-tape -lined carton. The Teflon tape facilitated carton removal and simulated a production-tooling surface. Each of the twenty mixes was cast to produce a rectangular- shaped block of 2.5-cm in width, 10-cm in length, and 12.5-cm in height, as depicted in Figure 1.

The front face (A) was arbitrarily identified as the surface that matched the label of the carton. The B surface is the top surface, the side from which the carton was cast.

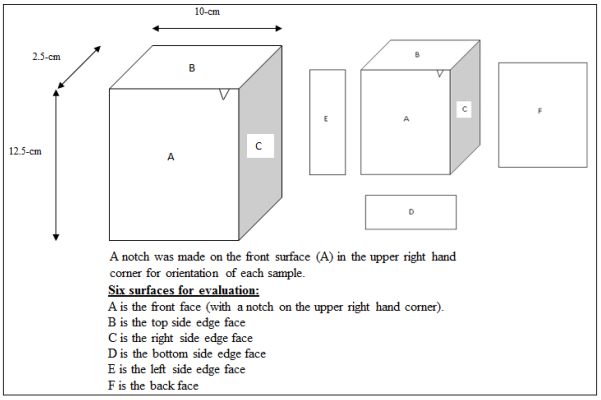


Figure 1: Sample Geometry and Sampling Locations

Upon vacuum-casting and cure of each of the twenty mixes, but prior to using microPHAZIR™ NIR platform for testing of mix constituency, Shore A hardness measurements were recorded instantaneously, at 10-second, and 15-second residence time. Table 3 summarizes shore A hardness measurements. Shore A hardness testing was performed on side B of all cast cartons. All propellant mixes experienced cure cycles of 145°F ± 5°F for 192 hours ± 24 hours. A mix would be removed from the cure oven if and only if shore A hardness conformed to minimum required specification.

Table 3: Shore A Hardness of the Twenty DOE Vacuum-Cast Mixes

Mix Number RBC1691-99-	Sample #	Oxidizer	Plasticizer	Antioxidant	Binder	TOTAL	Shore A Ambient Temperature		
							Instant	10 Sec	15 Sec
38 (Bay 8)	1	84	3.25	0.350	12.400	100	71	56	54
39 (Bay 5)	2	84	3.00	0.275	12.725	100	72	59	58
40 (Bay 8)	3	82	2.75	0.300	14.950	100	71	57	56
41 (Bay 5)	4	85	2.50	0.300	12.200	100	79	74	73
42 (Bay 8)	5	83	2.50	0.325	14.175	100	75	63	61
43 (Bay 5)	6	85	3.25	0.275	11.475	100	65	50	48
44 (Bay 8)	7	83	3.00	0.300	13.700	100	66	51	49
45 (Bay 5)	8	81	3.50	0.350	15.150	100	45	35	33
46 (Bay 8)	9	82	3.00	0.350	14.650	100	70	57	55
47 (Bay 5)	10	82	3.50	0.275	14.225	100	68	51	49
48 (Bay 8)	11	83	2.75	0.350	13.900	100	73	58	57
49 (Bay 5)	12	83	3.50	0.250	13.250	100	63	50	48
50 (Bay 8)	13	85	3.00	0.250	11.750	100	70	54	52
51 (Bay 5)	14	81	2.50	0.275	16.225	100	70	56	55
52 (Bay 8)	15	85	2.75	0.325	11.925	100	67	57	56
53 (Bay 5)	16	81	2.75	0.250	16.000	100	69	58	56
54 (Bay 8)	17	81	3.25	0.325	15.425	100	67	54	52
55 (Bay 5)	18	84	2.50	0.250	13.250	100	66	59	58
56 (Bay 8)	19	82	3.25	0.250	14.500	100	65	54	53
57 (Bay 5)	20	84	3.50	0.300	12.200	100	63	54	53

2.1. Instrumentation

2.1.1. The microPHAZIR™ (NIR) Platform

Near Infra-Red (NIR) spectroscopy (H.W. Siesler, Y. Ozaki, S. Kawata, and M. Heise (Eds), 2002) is a well-established technique, which has been widely used since the mid-1970s. Only recently has new technology permitted NIR systems to be miniaturized into truly handheld system. One of the most important products is the microPHAZIR™ NIR handheld platform. MicroPHAZIR™ NIR handheld platform is based on near-infrared spectroscopy ((H.W. Siesler). The near-infrared region, depicted in Figure 2 is located between the infrared and visible region with wavelengths that range from 800-900 nanometers to 2500 nanometers.

MicroPHAZIR™ NIR handheld platform was developed by Thermo Scientific and is based on vibrational spectroscopy (*microPhazir™* User Manual). All molecules perpetually rotate, move, and contort in a complex manner at temperatures above absolute zero. Vibrational spectroscopy probes these contortions (or vibrations) of a sample to determine the chemical functional groups present. A common type of vibrational spectroscopy is infrared (IR) absorption/reflectance. It relies on illumination of the sample with optical radiation to probe the molecular vibrations.

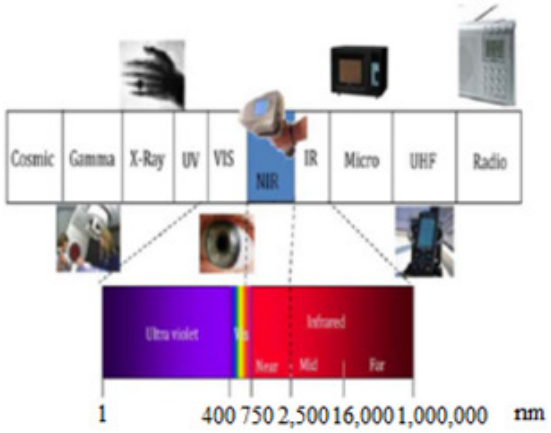


Figure 2: Near Infrared Region of the Light Spectrum

In NIR spectroscopy, the sample is illuminated with a broad spectrum of light in the near-infrared region and the transmission or reflection is recorded as a function of the frequency of the incident light. When the frequency of incident light equals the frequency of a specific molecular vibration, the sample tends to absorb some of the light. A material "fingerprint" results from recording the amount of light absorbed as a function of the wavelength (or frequency). The instrument is depicted in Figure 3. MicroPHAZIR™ NIR is a rugged handheld chemical identification unit designed for point-of-use applications, either in contact or analysis can be conducted through transparent bags and vials. This product allows the identification of chemicals and white powders using the principles of NIR spectroscopy. It is enclosed in a lightweight, rugged, resistant package. The microPHAZIR™ handheld contains a broadband NIR source, a Hadamard interferometer to separate the different wavelengths of light interacting with the sample, and a detector to collect the resulting energy.

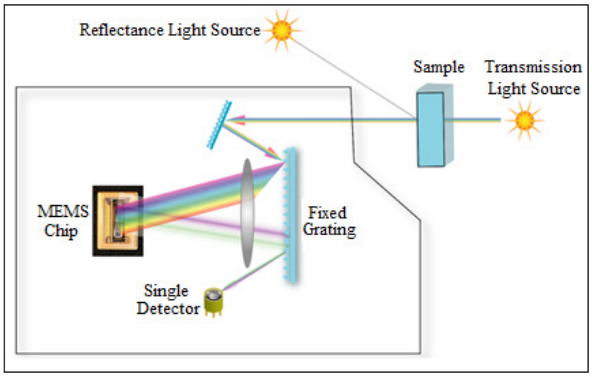


Figure 3: MicroPHAZIR™ NIR and Principle of Operation

2.1.2. Agilent 1100 HPLC Platform

Agilent 1100 Series system with different configurations comprises a vacuum degasser, isocratic pump, high-pressure binary pump, low-pressure quaternary pump, autosampler, thermostatted column compartment, variable wavelength detector and diode array detector. Key measurements are necessary to evaluate the performance of HPLC systems. Some characteristics are influenced by only one part of the system. For example, linearity, spectral resolution and detection limits are influenced mainly by the detector, delay volume and composition accuracy by the pump and carryover by the autosampler. In contrast, other characteristics such as baseline noise and precision of retention times and peak areas are influenced by the complete system. This note describes the following measurements:

I. Detector — baseline noise, drift, wander, linearity, spectral resolution, sensitivity.

II. Pump — composition accuracy, precision, ripple, precision of retention times, delay volume.

III. Column compartment — temperature stability.

IV. Autosampler — precision of peak areas, linearity, carry-over.

2.2. Chemical Health Management (CHM)

2.2.1. NIR Measurements/Data Collection

The primary measured constituents of CL-01 solid rocket propellant are listed herein:

I. Agerite White anti-oxidant/stabilizer.

II. Ammonium perchlorate (AP) Oxidizer.

III. Dioctyl Sebecate (DOS) plasticizer.

IV. Hydroxyl-terminated polybutadiene (HTPB) binder, and is determined by difference between the sum of the primary constituents in (I), (II), and (III) and 100%.

Measurements were performed on two platforms: microPHAZIR™ NIR handheld platform and Agilent 1100 High-Performance Liquid Chromatograph (HPLC) platform. In the case of microPHAZIR™ NIR handheld platform, measurements were performed and recorded on each of the six faces of each of the twenty rectangular blocks, as depicted in Figure 4.

Upon manufacture and vacuum casting of each propellant mix and prior to performing measurements, 3-mm of the binder-rich surface of each cast block is peeled-off and removed from the surface, exposing the homogeneous material.

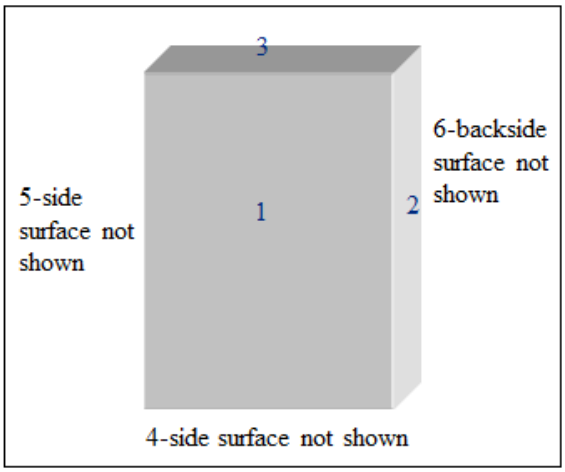


Figure 4: Vacuum-Cast Propellant Blocks, Depicting six Measured Surfaces

All 20 sample blocks were measured at ATK Launch Systems on June 26, 2013. The instrument used was microPHAZIR-GP Probe (part number 800-00259-01) with microPHAZIR Fiber Optic Probe Accessory (part number 810-01351-01). The instrument serial number was 2575 shown in Figure 3. The optical fiber probe attachment is depicted in Figure 5. The overall platform assembly is depicted in Figure 6 and Figure 7. Prior to taking measurements, microPHAZIR™ NIR platform was turned on and allowed to warm up for five minutes. A self-test pass was verified to ensure the unit was working properly.

Figure 5: Optical Fiber Probe

Samples were measured in sample number order starting with sample 1 (ATK mix number RBC1691-99-38) and ending with sample 20 (ATK mix number RBC1691-99-57). Each sample was measured in quadruplet and consecutively on each of its six faces/sides. The four measurements per face/side were collected at different positions. For each measurement, the tip of the optical fiber probe was placed in contact with the sample and perpendicular to the sample face so that the probe tip was flat against the sample surface. The instrument trigger was then activated to initiate an approximately three-second scan. Prior to collecting the four measurements per face, a

background measurement was taken of the built-in reflectance standard. The measurement technique is shown in Figure 6 and Figure 7.

Sample-averaging was performed in MATLAB. All other analyses were performed in Thermo Method Generator (TMG), version 4.0.1.0 (*microPhazir*™ User Manual).

Figure 6: Sample Side Measurement Technique

Figure 7: Sample Top Measurement Technique

2.2.2. Model Building (Schreyer, 2012)

2.2.2.1 Data Collection

CL-01 Spectral data using microPHAZIR™ NIR handheld platform followed best practices outlined by the platform manufacturer, as follows:

I. Obtain representative samples for the library.

A. Obtain realistic sample mixes that will form the library. These sample mixes should be representative of the CL-01 material that will be identified. No selectivity is implied for materials until the library is built and validated.

B. Measure samples, as illustrated in Figures 6 and 7. Perform measurements in triplicate.

C. Label all materials with name (Group ID or Method/Sample), and if appropriate reference value for PLS quantitative analysis.

D. Transfer all names into a ".csv" file, and then use this to populate "GroupID.csv" on the microPHAZIR™ "Config" directory.

II. Obtain reference values.

A. For quantitative analysis, the full range of measurement shall be included in the library. Models only are considered robust over the data range actually referenced.

B. Obtain replicate samples for at least 3 points over the measurement range.

C. For realistic model building (Schreyer, 2012), at least 10 reference values over the measurement range shall be obtained. As the size of the range increases, so should the reference values collected. Since samples may change over time, it is appropriate to collect the spectra from the same sample as the reference values are obtained from.

2.2.2.2 Spectral Generation

I. Pre-spectral collection

A. Prior to collecting spectra ensure that self-test performance qualification (PQ) has been performed.

B. Ensure that group identifications (group ids) are transferred into GroupID.csv.

C. Also ensure that the Group ID name is the correct name for the material and is present on the Collect screen on the microPHAZIR™.

II. Spectral collection

A. The minimum number of spectra collected for any library building is triplicate scans in 3 positions. Position the nose of microPHAZIR™ firmly against the material to be measured, as depicted in Figure 6 and Figure 7 (left), and take triplicate scans of the material without moving the sample. This will give information about instrument variability. Repeat twice.

B. Repeat measurements for each side of the block.

C. Repeat steps (A) and (B) for each mix.

2.2.2.3 Spectral Evaluation

I. Initial spectral evaluation.

A. Load the collected data into Method Generator

B. Ensure that there are no data which show absorbance (y-axis) past 3.

C. Observe if there are any noisy spectra, especially at high absorbance. If so, delete them. These usually arise if the trigger was pressed either without a sample in front or if sample is inadequate.

D. Highlight each group to make sure that all spectra look similar in the same group. Any obvious single outliers may be deleted. The best scenario is when the triplicate scans are right on top of each other, and there is little difference between positional scans. However, as long as the positional replicates appear similar and are close together, this is adequate. If one position is obviously off from the others, keep it, but watch to see if it affects the final results.

E. Delete any spectra where there was awareness of probable mistake in measurement. Do not delete scans just to make everything pretty. Deviations from the norm could be due to actual inherent sample differences and will need to become part of the model.

F. Reference values must be inputted at this time, using the Edit Y-value option.

G. Save the final edited data.

II. Method generation

A. Progress through the standard preprocessing options, and then evaluate the model using Spectral Match.

B. Adequate separation should be observed between samples. There should be a gap between the colors associated with one group and the next closest color of the nearest group.

C. Save the model if the model is acceptable.

D. Load the data files onto the microPHAZIR™ to test the model.

III. Method validation

A. Load a set of spectra into method generator (MG). For true method validation these should be unique spectra, not used in library building.

B. Select Model |Model validation. Browse to locate the application. Press OK

C. A panel will open with the validation results. It will be sorted by sample groups. Therefore it is very important that the GroupID of new spectra be identical to the GroupID of the library spectra. Otherwise a No ID label will be inserted.

D. The results show number of mismatches, false positives/false negatives, and then the full results of the model validation for each material. It will list the top 3 matches returned and their associated correlation coefficients.

E. The results can be saved as a ".csv" file by selecting File | Save all

2.3. Benchtop HPLC/IC Measurements/Data Collection

Prior to conducting measurements, approximately 3-mm of the surface of each of the 20 cast samples was removed. This process is often performed on freshly manufactured mixes because the first 3-mm of a cast composite propellant is often binder-rich. To maintain consistency in measurements, the binder- rich region was removed with a special cutting tool.

Following microPHAZIR™ platform measurements, 0.5- to 1-gram weight samples were removed from the measured regions and analyzed using HPLC and IC to determine the following (Mattos et al., 2004):

a. AP oxidizer content.
b. Agerite White stabilizer/antioxidant content.
c. DOS plasticizer content.
d. HTPB binder content (by difference).

HPLC and IC analyses of samples removed from the six surfaces of each of the twenty mixes are summarized in the results in section 3. For HPLC, samples were extracted overnight at a level of 50 mg/mL in stabilized tetrahydrofuran. Samples were prepared in triplicate. Sample extracts were analyzed using an HP1090 HPLC equipped with a C8 column and a diode-array detector. Approximately 200 milligrams were used for each sample preparation. A sample portion was cut with a razor blade into several pieces to facilitate extraction.

The antioxidant was identified by HPLC analysis (Urbanski et al, 1977) using a standard for identification. The antioxidant is often associated with the pre-polymer/binder matrix. The plasticizer was also measured using HPLC analysis (Urbanski et al, 1977).

IC samples were prepared by extracting 100 mg of propellant in 200 mg of deionized water. To facilitate the complete extraction of the AP from the propellant matrix, the propellant was leached for at least seven days in the water under ambient conditions. This may be a conservative amount of leach time, but evaluation after a 48-hour leach was shown to be inadequate. Exact weights of both the propellant sample and deionized water were recorded to at least four significant figures and used in the calculation. The chromatographic conditions of the analysis were performed as follows:

- Instrument: Dionex 500 IC System 4 with anion suppressed/conductivity detector
- Column: IonPac® AG4 guard column (4X50 mm)
- Eluent: 40 mM NaOH
- Flow rate: 1.2 ml/min
- Run time: 1.0 minute
- Elution time: 0.6 minutes
- Injection volume.: 10µLSRS Setting: 100 mA

- Calibration: bracketed
- Standards: 400.0, 430.0 and 460.0 ppm AP

2.4. Structural Health Management/Data Collection

Shore A hardness testing was performed on side B of all cast blocks, as depicted in Figure 1. Shore A hardness measurements were recorded at three residence times: instantaneous, 10-second, and 15-second residence times.

Measured values for shore A hardness were used to calculate elastic modulus, tensile strength, and percent strain for each propellant mix. Values were correlated to the specifications for CL-01. The primary goal of this technique is to validate mechanical integrity of the propellant real-time and nondestructively using shore A hardness measurement techniques. The approach would be utilized in conjunction with microPHAZIR™ handheld platform to determine structural as well as chemical health of CL-01 propellant and other composite propellants and explosives.

The following semi empirical formulae were used.in calculating tensile stress, elastic modulus, and percent strain, respectively.

For Tensile Stress (TS), using 15-second Shore A hardness measurements, stress was calculated in equation 1 (Shore (Durometer) Hardness) as follows:

$$TS = 0.0423\ (S_A)^{1.2799} \quad (1)$$

Where TS is tensile stress, in MPa, and S_A is Shore A hardness.

For Elastic Modulus E, using 15-second Shore A hardness measurements, E was calculated in equation 2 (A.N. Gent, 1958) as follows:

$$E = 0.0981\ (56 + 7.66S)/0.137505\ (254 - 2.54\ S_A) \quad (2)$$

Where E is elastic modulus in MPa.

Percent strain was calculated, in accordance with Hook's law, as the ratio of stress, in equation (1) to elastic modulus, in equation (2).

3. RESULTS

3.1. NIR Spectra

Figure 8 depicts unprocessed (unfiltered) spectra for each of the twenty CL-01 propellant samples. Data sets were collected using microPHAZIR™ NIR handheld platform. Almost all spectra were visually the same; with very few outliers, which is a normal trend. For example, the spectra in light blue have obvious spectral artifact at the high end at the wavelength end.

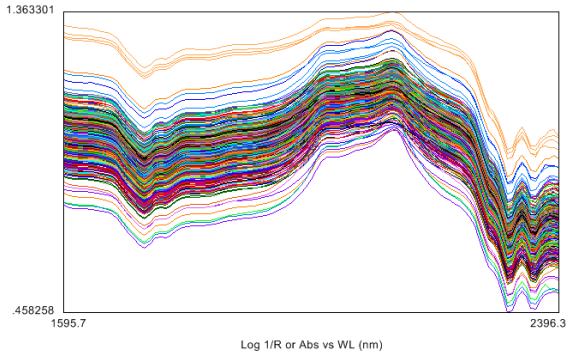


Figure 8: Unfiltered NIR Spectra (Absorbance vs. Wavelength)

These gross outlier spectra were removed (filtered) from the data sets (depicted in Figure 9). A total of 8 out of 480 (1.7 percent) spectra were removed as outliers leaving 472 spectra for use in the calibration and algorithm development models.

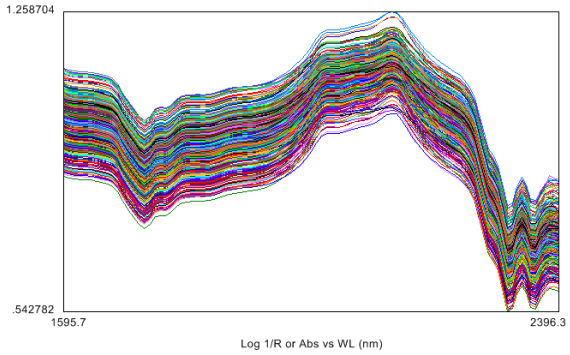


Figure 9: Filtered NIR Spectra (Absorbance vs. Wavelength)

For comparison, CL-01 propellant spectra and PBX(AF)-108 (S. Daoud, M. J. Villeburn, K. D. Bailey, G. Kinloch, L. Biegert, and C. Gardner, 2013) NIR spectra were plotted in Figure 10. It is observed that on average CL-01 spectra have approximately an absorbance near 1 and PBX have an absorbance near 0.4 (C. Gardner, and S. Schreyer 2013). This absorbance difference of 0.6 is translated to a $10^{-0.6}$, a factor of 0.25 of less light returned from CL-01 samples. This factor of four of less light is due to the gray color of CL-01 sample due to the presence of the opacifier in the formula. In general, the less light returned to the detector (due to higher light absorbance by the sample), the noisier the spectra.

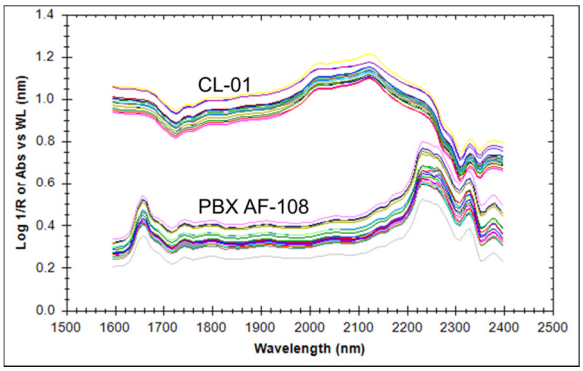


Figure 10: Comparison of Spectral Readings of CL-01 Propellant vs. PBX(AF)-108 Explosive

3.2. Spectral Preprocessing

To remove nuisance/noise variations from the NIR spectra before generating the calibrations models for algorithm development, the sample-averaged spectra were first filtered using Savitsky-Golay 9-point, 2nd-order, 2nd-derivative filtering technique. They were then range normalized so that each spectrum has zero mean and unity standard deviation.

Analysis was performed over the spectral range 1595–2250 nm, and this is the spectral range used for calibrations. Plots of the preprocessed spectra for each primary constituent are depicted in Figure 11.

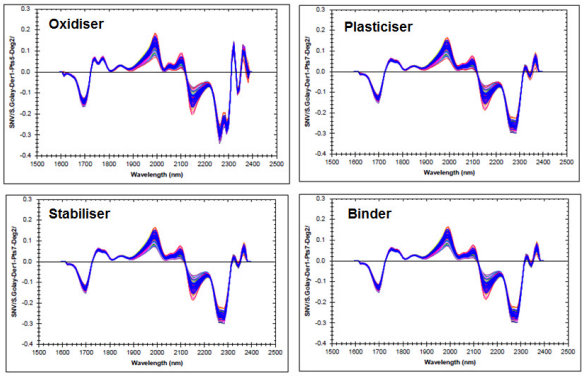


Figure 11: Sample Averaged, Pre-processed Spectra for the Four Primary Constituents of CL-01 Propellant

3.3. Design Points vs. microPHAZIR ™ NIR Readings

Test sets collected with microPHAZIR™ NIR handheld platform for all twenty samples were analyzed using Thermo Method Generator (TMG) partial least square (PLS) analysis software and are listed in Table 4. Upon reduction and analysis of the data, findings indicated near identical readings between those measured using microPHAZIR™ NIR and those measured using benchtop HPLC/IC instruments.

Table 4: microPHAZIR™ NIR Measurements of Primary Constituents

Mix Number RBC1691-99-	Sample #	% Oxidizer	% Plasticizer	% Stabilizer	% Binder
38	1	83.71	3.15	0.29	12.38
39	2	83.88	2.83	0.23	13.11
40	3	81.89	2.69	0.29	15.27
41	4	84.85	2.45	0.18	12.48
42	5	83.39	2.68	0.28	13.99
43	6	85.29	3.05	0.13	11.57
44	7	82.05	2.64	0.24	14.90
45	8	80.81	3.36	0.25	15.67
46	9	82.25	2.59	0.28	14.40
47	10	82.94	2.98	0.21	13.79
48	11	83.12	2.58	0.24	14.02
49	12	83.18	3.15	0.17	13.61
50	13	85.06	2.97	0.17	12.30
51	14	80.83	2.60	0.21	16.80
52	15	83.76	3.05	0.24	13.19
53	16	80.13	2.75	0.20	17.01
54	17	80.44	2.95	0.23	16.34
55	18	82.70	2.66	0.15	14.08
56	19	81.35	2.96	0.21	15.32
57	20	82.70	3.58	0.16	13.35

The plotted spectra of Figure 12 through Figure 15 for the oxidizer, plasticizer, stabilizer, and binder respectively are the sample-averaged results of the measured readings using microPHAZIR™ NIR (plotted on the Y-axis) versus design values, plotted on the X-axis, which were those of the individual DOE design sets listed.in Table 1.

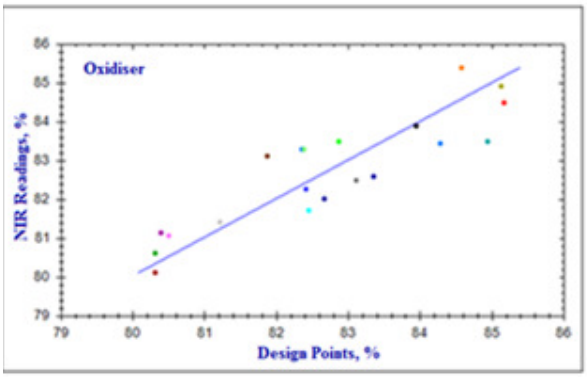


Figure 12: NIR Readings vs. DOE Design Points for the Oxidiser

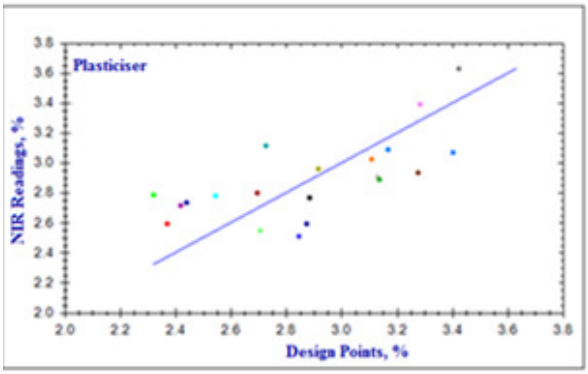


Figure 13: NIR Readings vs. DOE Design Points for the Plasticiser

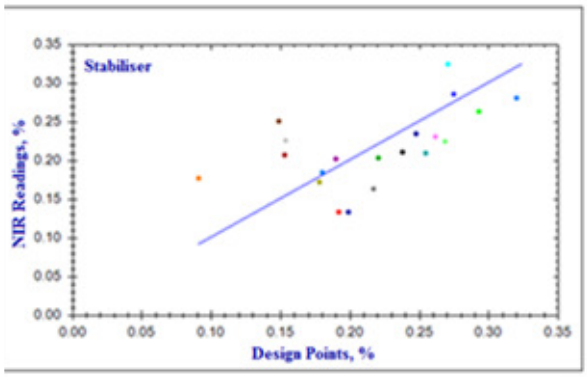


Figure 14: NIR Readings vs. DOE Design Points for the Stabiliser

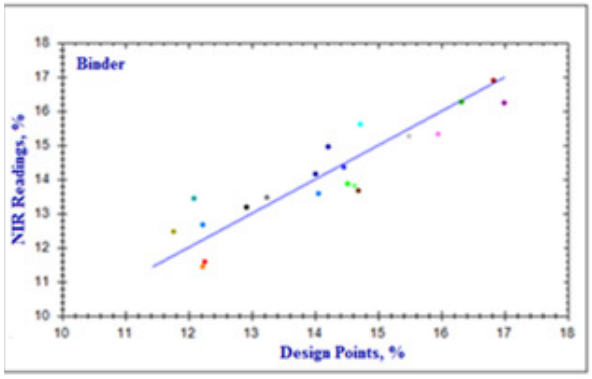


Figure 15: NIR Readings vs. DOE Design Points for the Binder

3.4. DOE Benchtop Analyses

Table 5 summarizes compositional results for each of the primary constituents in each of the 20 DOE sample mixes.

As illustrated in Figures 12 through 15 and Table 6, NIR values compare precisely well and within the allowable margin of error, to those measured using benchtop HPLC/IC and to DOE design sets of Table 1.

Table 5: HPLC Measurements of Primary Constituents

Mix Number RBC1691-99-	DOE Sample #	% Oxidizer	% Plasticizer	% Stabilizer	% Binder
38	1	84.29	3.17	0.32	12.22
39	2	83.95	2.89	0.24	12.92
40	3	82.46	2.55	0.27	14.72
41	4	85.17	2.37	0.19	12.26
42	5	82.87	2.32	0.29	14.51
43	6	84.58	3.11	0.09	12.22
44	7	82.67	2.88	0.25	14.21
45	8	80.51	3.29	0.26	15.94
46	9	82.42	2.85	0.27	14.46
47	10	81.88	3.28	0.15	14.69
48	11	82.39	2.71	0.27	14.63
49	12	82.36	3.40	0.18	14.06
50	13	85.13	2.92	0.18	11.77
51	14	80.40	2.42	0.19	16.99
52	15	84.94	2.73	0.25	12.08
53	16	80.32	2.70	0.15	16.83
54	17	80.32	3.14	0.22	16.32
55	18	83.35	2.44	0.20	14.01
56	19	81.22	3.13	0.15	15.49
57	20	83.12	3.42	0.22	13.24

Table 6: Comparison of Benchtop vs. Handheld Measurements of Primary Constituents

Mix Number	Sample	% Oxidizer		% Plasticizer		% Stabilizer		% Binder	
RBC1691-99-	#	HPLC	NIR	HPLC	NIR	HPLC	NIR	HPLC	NIR
38	1	84.29	83.71	3.17	3.15	0.32	0.29	12.22	12.38
39	2	83.95	83.88	2.89	2.83	0.24	0.23	12.92	13.11
40	3	82.46	81.89	2.55	2.69	0.27	0.29	14.72	15.27
41	4	85.17	84.85	2.37	2.45	0.19	0.18	12.26	12.48
42	5	82.87	83.39	2.32	2.68	0.29	0.28	14.51	13.99
43	6	84.58	85.29	3.11	3.05	0.09	0.13	12.22	11.57
44	7	82.67	82.05	2.88	2.64	0.25	0.24	14.21	14.90
45	8	80.51	80.81	3.29	3.36	0.26	0.25	15.94	15.67
46	9	82.42	82.25	2.85	2.59	0.27	0.28	14.46	14.40
47	10	81.88	82.94	3.28	2.98	0.15	0.21	14.69	13.79
48	11	82.39	83.12	2.71	2.58	0.27	0.24	14.63	14.02
49	12	82.36	83.18	3.40	3.15	0.18	0.17	14.06	13.61
50	13	85.13	85.06	2.92	2.97	0.18	0.17	11.77	12.30
51	14	80.40	80.83	2.42	2.60	0.19	0.21	16.99	16.80
52	15	84.94	83.76	2.73	3.05	0.25	0.24	12.08	13.19
53	16	80.32	80.13	2.70	2.75	0.15	0.20	16.83	17.01
54	17	80.32	80.44	3.14	2.95	0.22	0.23	16.32	16.34
55	18	83.35	82.70	2.44	2.66	0.20	0.15	14.01	14.08
56	19	81.22	81.35	3.13	2.96	0.15	0.21	15.49	15.32
57	20	83.12	82.70	3.42	3.58	0.22	0.16	13.24	13.35

Plots of primary constituents of CL-01 which were measured using microPHAZIR™ NIR, and HPLC/IC are depicted in Figures 16 through 19 and are correlated to actual DOE design sets. In the figures, only the stabilizer shows noticeable deviation from the design sets, primarily because the sum of Agerite white oxidation products and AO2246 antioxidant, often associated with Agerite white, were both excluded from PLS analysis. Those two components have no effect on stability of the propellant and are quantitatively negligible (amounting to approximately 5%) in comparison to the total plasticizer content.

As a result, this effort confirmed the ability to use the microPHAZIR™ NIR miniature handheld platform as a non-destructive means for determining primary chemical constituents of solid rocket propellants, and was validated via benchtop measurements of HPLC and IC instrumentation.

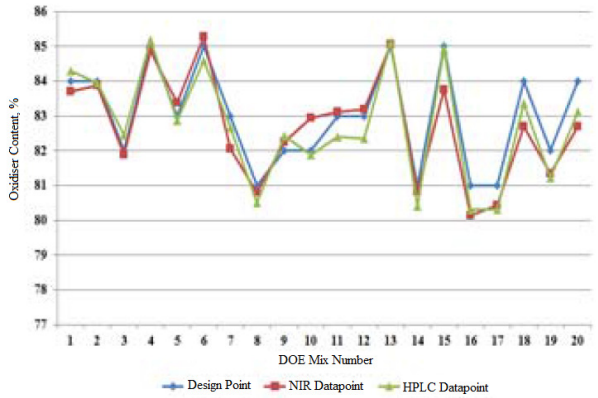


Figure 16: Comparison of NIR VS. HPLC Measurements to Actual Design Sets for Oxidiser Content

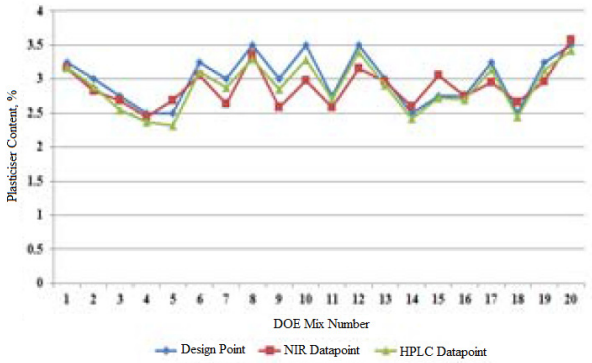


Figure 17: Comparison of NIR VS. HPLC Measurements to Actual Design Sets for Plasticiser Content

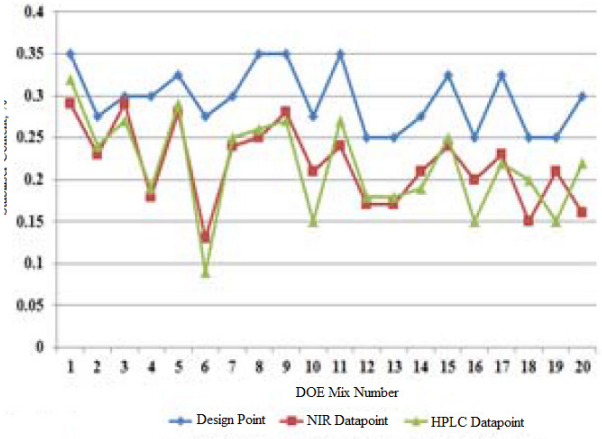


Figure 18: Comparison of NIR VS. HPLC Measurements to Actual Design Sets for Stabiliser Content

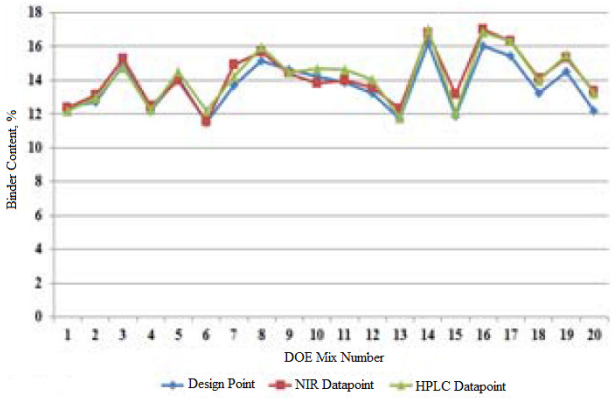


Figure 19: Comparison of NIR VS. HPLC Measurements to Actual Design Sets for Binder Content

3.5. Structural Health Management

Three critical elements define prognostics and health management as a real-time nondestructive test technique. The first element is electrical health management (EHM), and primarily comprises built-in test (BIT). This field is mature and has been used for over five decades. The second element is chemical health management (CHM), as a real-time nondestructive test technique. This field has been under development for the past decade and is only gaining grounds with the recent work of RMS. This work is described in this white paper and in an earlier white paper published at the International Journal of Prognostics and Health Management, in October of 2013. The third element is structural health management (SHM). Structural health management is a real-time nondestructive test technique comprising two sub-elements. The first is real-time nondestructive radiographic x-ray technique, which sheds light on incipient structural failures associated with cracks and crack propagation or delaminations at the propellant-liner-interface (PLI). This element is also mature and has existed for many decades, henceforth is not a topic of discussion in this white paper. However it is an integral sub-element of RMS structural health management. The second and most important sub-element of structural health management is the collection of shore A hardness data followed by manipulation of the data to yield results on propellant elastic modulus (E), tensile stress, and strain. This technique is nondestructive and is an integral sub-element of structural health management, which is an integral element of PHM. Upon collection of shore-A hardness data, elastic modulus is derived using equation (2), defined as Gent semi empirical formula. Shore-A hardness results are also used, as described in equation (1) to calculate tensile stress (TS). Strain is then calculated using the general equation described below:

$$\text{Modulus (E)} = \text{Stress/Strain} \quad (3)$$

Table 7 lists instantaneous, 10-second and 15-second shore-hardness for each of the twenty cast sample mixes of the DOE design sets. The table also lists the calculated moduli for each mix and compares them to maximum and minimum specification. It is important to keep in mind that elastic modulus is calculated from the 15-second shore-A hardness data. Fifteen-second shore-A hardness data are most reliable because the Durometer indenter is allowed sufficient time to penetrate the polymer-based composite material and henceforth provides more accurate and realistic values about the physical nature of the material elasticity.

In Table 7, only sample mix 4 had a modulus value that exceeded maximum specification of 750 psi. This is an expected result considering that mix 4 comprised the highest amount of oxidizer and lowest amount of plasticizer.

Table 7: Calculated Mechanical Data for CL-01 Propellant Using Shore Hardness Measurements

ATK CL-01 Mix Number	DOE Sample #	Shore A Hardness			Calculated Values from Shore A Hardness			Specifications		
		Inst.	10-S	15S	Modulus, Psi	Tensile Strength Psi	Strain, %	Modulus, Psi (Min. Max.)	Strain, % (Min.)	Tensile Strength, Psi (Min.)
38 (Bay 6)	1	71	56	54	453.9	1011.9	240.3			
39 (Bay 5)	2	72	59	58	483.3	1108.8	228.5			
40 (Bay 6)	3	71	57	56	469.1	1060.1	236.1			
41 (Bay 5)	4	79	74	73	928.2	1488.3	160.3			
42 (Bay 6)	5	75	63	61	569.3	1182.7	216.5			
43 (Bay 6)	6	65	50	48	331.9	870.3	262.2			
44 (Bay 6)	7	66	51	49	344.5	888.5	258.3			
45 (Bay 6)	8	45	35	33	187.8	538.7	286.9			
46 (Bay 6)	9	70	57	55	493.9	1035.9	230.9			
47 (Bay 5)	10	66	51	49	344.5	888.5	258.3	750 170	40	72
48 (Bay 6)	11	73	58	57	466.8	1094.4	232.3			
49 (Bay 6)	12	63	50	48	331.9	870.3	262.2			
50 (Bay 6)	13	70	54	52	389.0	994.1	255.2			
51 (Bay 5)	14	70	56	55	493.9	1035.9	230.9			
52 (Bay 6)	15	67	57	56	469.1	1060.1	236.1			
53 (Bay 6)	16	69	58	56	469.1	1060.1	236.1			
54 (Bay 6)	17	67	54	52	389.0	994.1	255.2			
55 (Bay 5)	18	66	59	58	483.3	1108.8	228.5			
56 (Bay 6)	19	65	54	53	400.3	947.9	246.8			
57 (Bay 5)	20	63	54	53	400.3	735.76	273.8			

Validation of this structural test technique is ongoing, and recent results collected from actual baseline (time t = 0) solid rocket propellant, as well as the same propellant under accelerated-aged conditions (time t = 1, 3, and 6 month) have indicated excellent correlation between specifications for modulus, stress, and strain and data which were calculated from measured shore-A hardness. The technique of using Shore A and Shore D hardness will be adopted as an integral means of real-time nondestructive test technique in combination with x-ray radiography, for structural health management (SHM). This technique in combination with microPHAZIR™ NIR will be an integral part of prognostics and health management (PHM), as a real-time NDT/NDE test technique to future surveillance of solid rocket propellants and warhead explosives.

4. CONCLUSION

Datasets from both microPHAZIR™ NIR handheld platform and Agilent 1100 (Performance Characteristics) high-performance liquid chromatography (HPLC)/ion chromatography (IC) platforms were precisely similar and representative of the constituents of CL-01 solid rocket propellant. In the case of microPHAZIR™ NIR handheld platform, dataset indicated excellent consistency and stability across the full datasets while at the same time closely representative of the results collected using Agilent 1100 high-performance liquid chromatography (HPLC)/ion chromatography (IC) platform.

The D-optimal full-factorial design of experiment (DOE) was successful in generating an algorithm for use in microPHAZIR™ NIR handheld platform for use in real-time quantitative determination of primary chemical constituents of CL-01 solid rocket propellant. Therefore, use of microPHAZIR™ NIR handheld platform for real-time non-destructive chemical quantification solid rocket propellants is a valid chemical health management (CHM) test technique, which alleviates the drawbacks of chemical waste and solid residue generation.

Of notable importance in this work is the concurrent success of using Shore-A hardness as a real-time nondestructive test technique for determining mechanical properties of the propellant, henceforth structurally monitors health of the propellant matrix (SHM).

Therefore, the combination of chemical and structural health management of the solid rocket propellant was successfully demonstrated in this work as a primary means of real-time prognostics and health management (PHM) technique for energetic and inert composite polymer based materials.

NOMENCLATURE

AF	Air Force
AOP	Allied Ordnance Publication
AP	Ammonium Perchlorate
ATK	Alliant Techsystems
BTCA	Breakdown, Test and Criticality Analysis
CAD	Cartridge-Actuated Device
CHM	Chemical Health Management
CT	Computed Tomography
DOS	Dioctyl Sebecate
DoD	Department of Defence
DSTO	Defence Science and Technology Organization
EED	Electro-Explosive Device
HPLC	High-Performance Liquid Chromatography
IC	Ion Chromatography
LS	Launch Systems
MoD	Ministry of Defence

NATO North Atlantic Treaty Organization

NDE Non-Destructive Evaluation

NDT Non-Destructive Testing

NIR Near-Infrared

PAD Propellant-Actuated Device

PBX Plastic-Bonded Explosive

PHM Prognostics and Health Management

PLS Partial Least Square

SHM Structural Health Management

REFERENCES

S. Daoud, M. J. Villeburn, K. D. Bailey, G. Kinloch, L. Biegert, and C. Gardner, 2013. "Determination of Primary Chemical Constituents of PBX(AF)-108 Warhead Explosive using *microPHAZIR™* Near Infrared (NIR) Handheld Platform", Annual Conference of the Prognostics and Health Management Society, 2013.

C. Gardner, and S. Schreyer 2013. "*microPhazir™* CL-01 Solid Rocket Propellant Quantitative Results". Thermo Fisher Scientific, Tewksbury, MA 01887, USA.

C. Gardener, and M. Hargreaves, 2012. "Near Infrared Data Report for PBX(AF)-108 Warhead Explosive". Thermo Fisher Scientific, Tewksbury, MA 01887, USA.

S. Schreyer, 2012, Thermo Scientific Training Course Tutorial Series: "Building Quantitative (PLS-1) Models". Thermo Scientific, Tewksbury, MA 01887, USA.

S. Schreyer, 2012, "Thermo Scientific Best Practices for Collecting and Evaluating Spectra from *microPhazir™* NIR handheld platform". Thermo Scientific, Tewksbury, MA 01887, USA.

L. Biegert, and B. Cragun, 2013, "D-Optimal Design of Experiment for Qualification of *microPhazir™* Handheld NOR Platform on Experimental Rocket Motor Propellant", Final Report No. TR-034059. ATK Launch Systems, Aerospace Systems, Brigham City, UT 84302, USA.

G. Bocksteiner, and D.J. Whelan, *November 1995,* DSTO-TR-0228: "The Effect of Ageing on PBXW-115(Aust.) PBXN-103 and PBXN-105". *Department of Defence, Defence Science* and Technology Organization (DSTO).

Mattos et al., 2004, "Determination of the HMX and RDX Content in Synthesized Energetic Material by HPLC, FT-MIR, and FT-NIR Spectroscopies", Química Nova, Vol. 27, No. 4, pp. 540-544.

Urbanski et al, 1977, "Handbook of Analysis of measures of Synthetic Polymers and Plastics", John Wiley & Sons, New York, 494 p.

M. Blanco, and I. Villarroya (2002) NIR spectroscopy: "A rapid-response analytical tool". Trends in analytical chemistry 21:240-250.

H.W. Siesler, Y. Ozaki, S. Kawata, and M. Heise (Eds) (2002). Near Infrared Spectroscopy Principles, Instruments, Wiley-VCH.

microPhazir™ User Manual, Thermo Scientific Handheld Near-Infrared Analyzer. Thermo Fisher Scientific, Tewksbury, MA 01887, USA.

Performance Characteristics of the Agilent 1100 Series Modules and Systems for HPLC. Agilent Technologies, Publication Number 5965-1352E.

A.N. Gent (1958), On the relation between indentation hardness and Young's modulus, Institution of Rubber Industry – Transactions, 34, pp. 46–57.

"Shore (Durometer) Hardness Testing of Plastics". Retrieved 2006-07-22.

Online Normalization Algorithm for Engine Turbofan Monitoring

Jérôme Lacaille[1], Anastasios Bellas[2]

[1]*Snecma, 77550 Moissy-Cramayel, France*
jerome.lacaille@snecma.fr

[2]*SAMM, Université Panthéon-Sorbonne, 75013 Paris, France*
anastasios.bellas@malix.univ-paris1.fr

ABSTRACT

To understand the behavior of a turbofan engine, one first needs to deal with the variety of data acquisition contexts. Each time a set of measurements is acquired, and such set may account for tens of parameters, the aircraft evolves in a specific flight mode. A diagnostic of the engine behavior models the observations and tests if anything appears as expected. A model of the engine measurement vector may be very complex to produce and even more to deploy on board. The idea is to solve the problem locally on recurrent phases on which each single problem may be easier to answer. Civil flight missions are straightforward to decompose as they are very recurrent. It is more difficult with military missions and bench tests. Once a set of phases is defined, local regression models may be built. To solve nonlinearities a selection of computed variables is a good approach but such algorithm needs the definition of a stable set of recurrent phases and a very complex learning procedure that uses a huge amount of memory to deal with the high dimensionality of the problem. Such algorithm is very powerful but is not adapted for an online use. Our new solution does not require the a priori knowledge of recurrent phases; it learns recurrent contexts on the fly and adapts a small local regression model on a selected optimal subspace. The application of this algorithm seems to be efficient on long term flight trend monitoring and on real time test bench measurements. It solves the memory problem for calibration by an iterative autoadaptive procedure and suppress the need of preliminary computations of specific parameter as it auto-adapts itself with piecewise linear models.

1. INTRODUCTION

Turbofan engine abnormality diagnosis uses three steps: reduction of dependencies from the flight context (1), representation of the measurement in an adequate metric space suitable for classification and statistic testing (2) and finally identification of abnormal behavior (3) as represented on Figure 1 (next page). This work essentially deals with the first normalization step.

The current text focus on identification of flight phases to extract subsamples of temporal observations where the turbofan gross behavior may be explained by simple (eventually linear) models. This example is easy to visualize, but we also use the same algorithm on different applications. At component level we monitor the start system (Flandrois, Lacaille, Massé, & Ausloos, 2009; Lacaille, 2009), the fuel system and other turbofan components. Even to monitor bench test cells we look at vibration monitoring according to load parameters and lots of different other configurations (Lacaille & Gerez, 2011, 2012).

The first step of the algorithm is to get rid of acquisition context. This is mandatory because we need to compare similar events, observations corresponding to one unique and standard context. For this purpose we use a normalization algorithm (Figure 1, step 1). The classical method is to use a model of the engine observation measurements named endogenous parameters according to the flight context also referred as exogenous parameters (see Table 1 for a list of parameter examples). The residual between real endogenous parameters and the model results is then used as inputs to a scoring algorithm (Figure 1, step 2) which is essentially a statistical test that measures the likelihood of the current observation. The main problem is the construction of such residual. As the engine behavior is definitely nonlinear according to the flight measurements a suggestion is to cut the flight in recurrent phases: taxi, takeoff, climb, cruise, descent, etc. and models the behavior locally on those phases. However as such decomposition seems easy to build on civil mission it is a real challenge on

Jérôme Lacaille et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

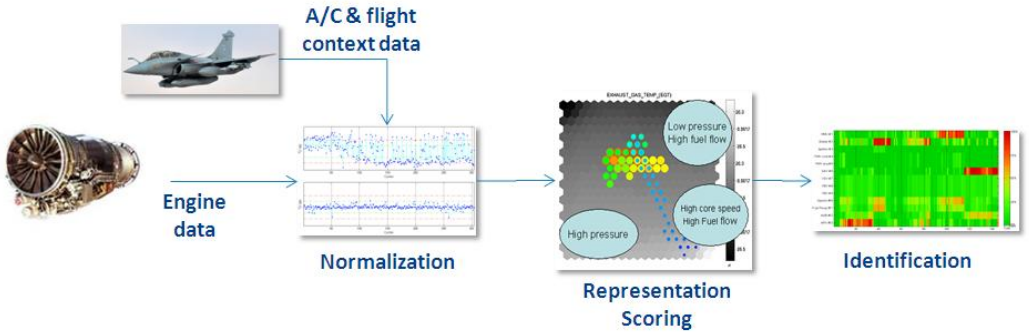


Figure 1 – The mains steps of any diagnostic application for aircraft engine monitoring.

military missions which all are different as well as for helicopter missions, business jets and event test bench tests.

Table 1 – Example of context information and endogenous measurements. Context parameters are mainly commands that describe the current engine use but also aircraft attitude. Endogenous measurements represent the observation we are really interested in to describe the engine behavior if the context was always the same during acquisition

Name	Description
Index information	
AC_ID	*Aircraft ID*
ESN	*Engine Serial Number*
FL_DATE	*Flight Date*
Context information	
TAT	*External temperature*
ALT	*Altitude*
AIE	*Anti Ice Engine*
AIW	*Anti Ice Wings*
BLD	*Bleed valve position*
ISOV	*ECS Isolation Valve Position*
VBV	*Variable Bleed Valve Position*
VSV	*Variable Stator Vane Position*
HPTACC	*High Pressure Turbine Active Clearance Control*
LPTACC	*Low Pressure Turbine Active Clearance Control*
RACC	*Rotor Active Clearance Control*
ECS	*Environmental Control System*
TLA	*Thrust Lever Angle*
N1	*Fan Speed*
XM	*Mach Number*
Endogenous measurements	
N2	*Core Speed*
FF	*Fuel Flow*
PS3	*Static pressure after compression*
T3	*Temperature after compression*
EGT	*Exhaust Gas Temperature*

Our first approaches uses manual extraction of flight phases for civil engines and a LASSO algorithm for the selection of pertinent analytical combinations of parameters to build the regression model and then a autoadaptive clustering method that uses a self-organizing map (SOM) to identify the different faults or behavior differences (Figure 1, step three "identification"). This work was presented in previous work (Côme, Cottrell, Verleysen, & Lacaille, 2010, 2011; Cottrell et al., 2009; Lacaille & Côme, 2011).

Even when flight mode identification of recurrent phases is clear, the normalization model that currently uses a LASSO regression algorithm needs a very huge amount of memory. The LASSO algorithm needs a matrix of the parameter measurements in memory: as an example the data for one engine from a set of 500 medium range flights with 100 parameters weight around 1.5 Gb when acquired at 1 Hz. Even this volume of data is not easily manageable with classical tools and standard algebraic operations such as singular values decomposition (SVD) which is the base tool in linear compression. Hence it is only possible to calibrate this model on ground on a subsample of data we may download from a small subset of aircrafts which owners (the airlines or military) let us have access to their digital flight data recorders (DFDR, the black boxes). The resulting model transferred on each engine is finally a general approximation. It misses the specificity of each engine or event the particular way each company and pilot operates its aircrafts.

2. STATISTIC MIXTURE MODEL

To solve our normalization problem iteratively with not too much memory resources involved we used a mixture of probabilistic principal component analysis (MPPCA) model. Such model is an extension of the classical PCA which goal is to extract a reduced number of dimensions on which the data may be explained. The reduction of dimension enables the computation of meaningful distances[1] and allows the

[1] Distances are needed to compute a score based on the likelihood of the difference between observation and model estimation. In high dimension,

computation of scores. However if the general behavior of observations is not linear a classical PCA algorithm will fail. A nice solution is to make the hypothesis that in each flight mode, the local behavior of the engine may have a linear representation. The MPPCA algorithm will use EM (expectation/maximization) optimization scheme to identify the clusters and to build the local projections.

We consider that we dispose of a datastream $D = \{\mathbf{x}_1, \dots \mathbf{x}_N\} \in R^p$, where the $\mathbf{x}$ are independent realizations of a random vector $X \in R^p$. In addition, $\{z_1, \dots, z_N\}$ are assumed to be independent realizations of an unobserved (latent) random variable Z with values in $\{1, \dots, K\}$ (there exists K different modes.) The MPPCA model assumes that the observed random vector $X \in R^p$ is, conditionally to Z, linked to a d-dimensional latent random vector $Y \in R^d$ through a linear transformation of the form:

$$X_{|Z=k} = U_k Y + \boldsymbol{\mu}_k + \epsilon \tag{1}$$

where U_k is the $p \times d$ orthogonal transformation matrix, $\mu_k \in R^p$ is the mean vector of the k-th factor analyzer and $\epsilon \in R^p$ is a noise term. The dimension d of the latent vector is such that $d < p$ and assumed to be known (Figure 2 below shows an illustration of K=2 d=2D subspaces in a p=3D domain).

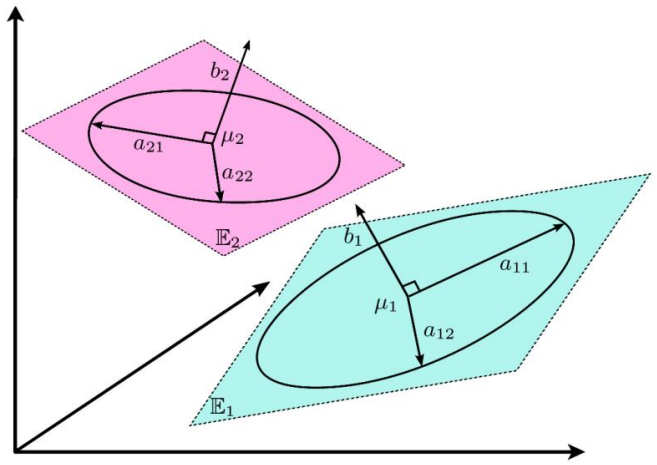


Figure 2 – Illustration of two Gaussian d=2 subspaces in a main p=3 dimension space.

Moreover, ϵ is assumed to be, conditionally to Z, a centered Gaussian noise term with a diagonal covariance matrix $b_k I_p$:

$$\epsilon_{|Z=k} = \mathcal{N}(0, b_k I_p). \tag{2}$$

Besides, the unobserved latent factor $Y \in R^d$ is assumed to be, conditionally to Z, distributed according to a Gaussian density function such as:

$$Y_{|Z=k} = \mathcal{N}(0, I_d). \tag{3}$$

This implies that the conditional distribution of X is also Gaussian:

$$X_{|Y,Z=k} \sim \mathcal{N}(U_k Y + \boldsymbol{\mu}_k, b_k I_p) \tag{4}$$

and its marginal distribution is therefore a mixture of Gaussians:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) \tag{5}$$

where π_k is the mixture proportion for the k-th component, ϕ is the multivariate Gaussian density function

$$\begin{aligned} \phi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) = & \frac{1}{(2\pi)^{\frac{p}{2}} \det(\Sigma_k)^{\frac{1}{2}}} \times \\ & \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)' \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right) \end{aligned} \tag{6}$$

and $\Sigma_k = U_k' U_k + b_k I_p$.

In order to facilitate the description of our online inference procedure, let us slightly re-parameterize the above model. Let us first introduce the orthonormal transformation matrix Q_k which is such that its j-th column $q_{kj} = u_{kj} / \|u_{kj}\|$ where u_{kj} is the corresponding column of U_k. If the transformation matrix Q_k is orthonormal, it is then necessary to report the variance of the latent factor within the distribution of the latent factor.

We therefore now assume that $Y_{|Z=k} = \mathcal{N}(0, \Delta_k)$ where $\Delta_k = \text{diag}(\lambda_{k1}, \dots \lambda_{kd})$. The marginal distribution of X is then still a mixture of Gaussians but with covariance matrices $\Sigma_k = Q_k' \Delta_k Q_k + b_k I_p$. By denoting by $Q_k = [Q_k, R_k]$ the $p \times p$ matrix made of Q_k and an orthonormal complementary R_k, the projected covariance matrix $Q_k \Sigma_k Q_k'$ has the following form:

$$\begin{pmatrix} \begin{bmatrix} a_{k1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{kd} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} b_k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_k \end{bmatrix} \end{pmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} d \\ (p-d) \end{matrix}$$

where $a_{kj} = \lambda_{kj} + b_k$ and $a_{kj} > b_k$, for $k = 1, \dots K$ and $j = 1, \dots d$. With these notations, the mixture of PCA model is fully parameterized by the set of parameters $= (\theta_k)_{k=1 \dots K}$ with each $\theta_k = \{\pi_k, \mu_k, Q_k, (a_{kj})_{j=1 \dots d}, b_k\}$.

It can be shown (Bouveyron, Girard, & Schmid, 2007) that the MPPCA model is identifiable and its inference can be done using a simple EM algorithm. In particular, the update

distances lose their signification which is also known as the curse of high dimensions. We try to limit ourselves to a selected dimension smaller than 5.

formula in the M step for the orientation matrices Q_k and the variance parameters a_{kj} and b_k are as follows:

- the d columns of Q_k are estimated by the eigenvectors associated with the d largest eigenvalues of the empirical covariance matrix S_k of the k-th group,
- the empirical covariance matrix of the k-th group is $S_k = \frac{1}{n}\sum_{i=1\dots n} t_{ik}(x_i - \mu_k)(x_i - \mu_k)'$ where at each current step $t_{ik} = P(Z = k|\mathbf{x}_i; \theta)$.
- a_{kj} is estimated by the j-th largest eigenvalues of S_k,
- b_k is estimated by the residual variance $b_k = \frac{1}{p-d}\left(\mathrm{tr}(S_k) - \sum_{j=1}^{d} a_{kj}\right)$.

In addition, these update formulas illustrate the strong link between MPPCA and the principal component analysis (PCA) method, since they both consider eigenvectors corresponding to the largest eigenvalues of the covariance matrix eigen decomposition.

3. ONLINE INFERENCE OF PARAMETERS

A standard way to estimate model parameters in parametric mixture models is to maximize the (observed) log-likelihood of the data.

$$L(\mathbf{x}; \theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \phi(\mathbf{x}_i, \theta_k) \tag{7}$$

Note that we prefer the log-likelihood over the likelihood, as it is much more convenient to work with the former from a mathematical point of view. The maximum likelihood method then proposes to estimate the parameters of the model θ by $\hat{\theta}_{MV} = \arg\max L(x; \theta)$.

As we saw earlier, complete data $\{(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), \dots, (\mathbf{x}_n, z_n)\}$ are composed of pairs of data $\mathbf{x}$ and class information z. The complete log-likelihood $L_c(\mathbf{x}, z; \theta)$ is the log-likelihood calculated from the complete data:

$$L_c(\mathbf{x}, z; \theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_k^{(i)} \log\left(\pi_k \phi(\mathbf{x}_i, \theta_k)\right) \tag{8}$$

Here, we have defined t as the indicator variable of the classes, so that if $z_i = k$ for a data sample i, then $t_k^{(i)} = 1$ and $t_k^{(j)} = 0, \forall j \neq k$.

In order to extend MPPCA to the online setting, we develop hereafter an online EM-based algorithm which incorporates a probabilistic version of the incremental PCA (Hall, Marshall, & Martin, 1998). We consider here a setting where data samples are arriving in an online manner and each data sample is being discarded after being processed (Bellas, Bouveyron, Cottrell, & Lacaille, 2013).

Let us assume that we initially have observed a dataset of n_0 data samples $(\mathbf{x}_1 \dots \mathbf{x}_{n_0}) \in R^p$ and that we have obtained an initial estimate $\hat{\theta}^{(n_0)}$ of these data. In practice, we obtain an initial estimation of the model parameters with a standard MPPCA iterative EM algorithm on this initial dataset. Let us set $n = n_0$ and consider the arrival of a new data sample $\mathbf{x}_{n+1} \in R^p$.

The objective is therefore to update the estimate of θ from the sole knowledge of $\hat{\theta}^{(n)}$ and $\mathbf{x}_{n+1}$. This is a two-step procedure which involves an expectation step (E-step) and a maximization step (M-step).

3.1. The E-step

Before updating the estimate of θ, it is necessary to compute the expectation of the complete log-likelihood $E\left(L_c(\mathbf{x}, z; \theta)\middle|\theta^{(n)}\right)$ conditionally to the current estimate $\hat{\theta}^{(n)}$.

This quantity will be maximized in the second step to obtain the new estimate $\hat{\theta}^{(n+1)}$ of θ. As with all mixture models, the computation of the conditional expectation of the complete log-likelihood reduces, in the context of the MPPCA model, to the computation of the probabilities $t_k^{(n+1)} = P\left(Z = k\middle|X = \mathbf{x}_{(n+1)}\right)$ that the new data sample belongs to the k-th mixture component (Figure 3). These probabilities can be computed as follows:

$$\begin{aligned} t_k^{(n+1)} &= \frac{\pi_k \phi\left(x_{n+1};\ \hat{\theta}_k^{(n)}\right)}{\sum_{l=1}^{K} \pi_l \phi\left(x_{n+1};\ \hat{\theta}_l^{(n)}\right)} \\ &= 1 \Big/ \sum_{l=1}^{K} \exp\left(\frac{1}{2}\left(\Gamma_k^{(n)}(x_{n+1}) - \Gamma_l^{(n)}(x_{n+1})\right)\right) \end{aligned} \tag{9}$$

where the classification function Γ_k has the following form:

$$\begin{aligned} \Gamma_k(x) = {} & \|\boldsymbol{\mu}_k - P_k(\mathbf{x})\|^2_{\mathcal{A}_k} + \frac{1}{b_k}\|\mathbf{x} - P_k(\mathbf{x})\|^2 \\ & + \sum_{j=1}^{d} \log(a_{kj}) + (p-d)\log(b_k) - 2\log(\pi_k) \end{aligned} \tag{10}$$

$$\text{with} \begin{cases} \|\mathbf{x}\|^2_{\mathrm{A_k}} = \mathbf{x}'\mathrm{A_k}\mathbf{x} \\ \mathrm{A_k} = \mathrm{Q_k}\Delta_k^{-1}\mathrm{Q_k}' \\ \mathrm{P_k}(\mathbf{x}) = \mathrm{Q_k}\mathrm{Q_k}'(\mathbf{x}\text{-}\boldsymbol{\mu}_k) + \boldsymbol{\mu}_k. \end{cases}$$

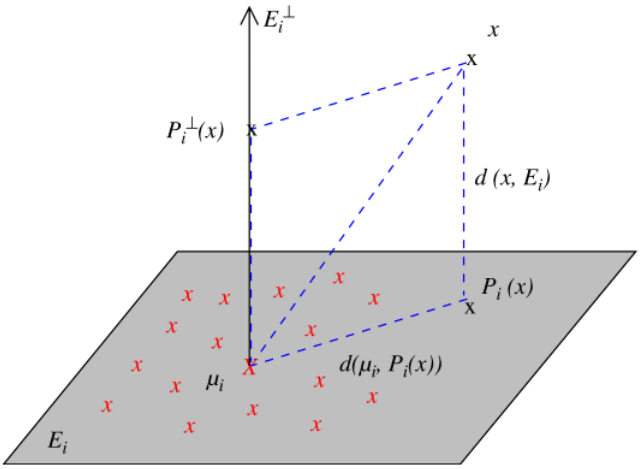


Figure 3 – Geometric interpretation of the probability that a sample belongs to a given class.

3.2. The M-step

Once the posterior probabilities $t_k^{(n+1)}$ have been computed, we update the model parameters so that they maximize $E\left(L_c(\mathbf{x}, z; \theta)\middle|\theta^{(n)}\right)$. In order to derive an online inference strategy which does not keep all past data samples, it is necessary to make use of the following approximation:

$$E\left(L_c(\mathbf{x}, z; \theta)\middle|\theta^{(n)}\right) \simeq E\left(L_c(\mathbf{x}, z; \theta)\middle|\theta^{(n-1)}\right) + \sum_{k=1}^{K} t_k^{(n+1)} \log\left(\pi_k \phi\left(\mathbf{x}_{n+1}; \theta_k^{(n)}\right)\right) \quad (11)$$

Then, it is straightforward to show that the update formulas for the mixture proportions π_k and the component means $\boldsymbol{\mu}_k$, for every component $k = 1 \dots K$, are:

$$\begin{aligned} \pi_k^{(n+1)} &= \pi_k^{(n)} + \tfrac{1}{n+1}\left(t_k^{(n+1)} - \pi_k^{(n)}\right), \\ \boldsymbol{\mu}_k^{(n+1)} &= \tfrac{1}{n_k^{(n+1)}}\left(n_k^{(n)}\boldsymbol{\mu}_k^{(n)} - t_k^{(n+1)}\mathbf{x}_{n+1}\right), \end{aligned} \quad (12)$$

where $n_k^{(n+1)} = n_k^{(n)} + t_k^{(n+1)}$ and $n = \sum_{k=1\dots K} n_k^{(n)}$.

We then want to estimate the parameters Q_k, a_{kj} and b_k, for $k = 1 \dots K$ and $j = 1 \dots d$. We have already seen that the maximization of $E\left(L_c(\mathbf{x}, z; \theta)\middle|\theta^{(n)}\right)$ with respect to these parameters is equivalent to the eigen decomposition of the empirical covariance matrix S_k for each component $k = 1 \dots K$. The problem that we seek to solve can be therefore stated as follows: having already calculated eigenvectors $Q_k^{(n)}$ and eigenvalues $\Lambda_k^{(n)}$ from the n first data samples, we want to update those parameters on the arrival of a (n+1)-th data sample. In particular, on the arrival of the new data sample $\mathbf{x}_{n+1}$, the new eigenproblem that we need to solve is:

$$\Sigma_k^{(n+1)} Q_k^{(n+1)} = Q_k^{(n+1)} \Lambda_k^{(n+1)} \quad (13)$$

where $\Lambda_k^{(n+1)} = diag\{\lambda_{k1} \dots \lambda_{kp}\}$ and this for $k = 1 \dots K$. To begin with, let us define:

$$\begin{aligned} g_k^{(n+1)} &= \left(Q_k^{(n)}\right)'\left(t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n)}\right) \\ h_k^{(n+1)} &= \left(t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n)}\right) - Q_k^{(n)} g_k^{(n+1)} \end{aligned} \quad (14)$$

where $g_k^{(n+1)}$ is the projection of the data sample on the subspace defined by the eigenvectors and $h_k^{(n+1)}$ is the residue of the retro-projection on the original space. With these notations, the new eigenvectors $Q_k^{(n+1)}$ correspond to a rotation of the old ones plus the unit residue vector $\tilde{h}_k^{(n+1)}$:

$$\tilde{h}_k^{(n+1)} = \begin{cases} \dfrac{h_k^{(n+1)}}{\left\|h_k^{(n+1)}\right\|_2}, & \text{if } \left\|h_k^{(n+1)}\right\|_2 \neq 0 \\ 0, \text{otherwise.} \end{cases} \quad (15)$$

and thus the new eigenvectors may be written:

$$Q_k^{(n+1)} = \left[Q_k^{(n)}, \tilde{h}_k^{(n+1)}\right] R_k^{(n+1)} \quad (16)$$

where $R_k^{(n+1)}$ is a rotation matrix of size (d+1)×(d+1). Note that $Q_k^{(n)}$ is a $p \times d$ matrix, since we have discarded the $p-d$ less significant eigenvalues. The new covariance matrix $\Sigma_k^{(n+1)}$ for the class k is given by:

$$\Sigma_k^{(n+1)} = \frac{n_k^{(n)}}{n_k^{(n+1)}}\Sigma_k^{(n)} + \frac{n_k^{(n)}}{\left(n_k^{(n+1)}\right)^2}\bar{\mathbf{x}}\bar{\mathbf{x}}' \quad (17)$$

where we have set $\bar{\mathbf{x}} = t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n+1)}$. Then, by substituting equations (16) and (17) into equation (13) we get[2]:

$$\left[Q_k^{(n)}, \tilde{h}_k\right]'\left(\frac{n_k^{(n)}}{n_k^{(n+1)}}\Sigma_k^{(n)} + \frac{n_k^{(n)}}{\left(n_k^{(n+1)}\right)^2}\bar{\mathbf{x}}\bar{\mathbf{x}}'\right)\left[Q_k^{(n)}, \tilde{h}_k\right] R_k^{(n+1)} = R_k^{(n+1)}\Lambda_k^{(n+1)} \quad (18)$$

The above problem can be written as:

$$\left(\frac{n_k^{(n)}}{n_k^{(n+1)}}\begin{bmatrix}\Lambda_k^{(n)} & 0 \\ 0 & 0\end{bmatrix} + \frac{n_k^{(n)}}{\left(n_k^{(n+1)}\right)^2}\begin{bmatrix}g_k g_k' & \gamma_k g_k \\ \gamma_k g_k' & \gamma_k^2\end{bmatrix}\right) R_k^{(n+1)} = R_k^{(n+1)}\Lambda_k^{(n+1)} \quad (19)$$

where we have set $\gamma_k^{(n+1)} = \tilde{h}_k^{(n+1)\prime}\bar{\mathbf{x}}$. The solution to this new eigenproblem yields the rotation matrix $R_k^{(n+1)}$ and the new eigenvalues $\Lambda_k^{(n+1)}$ directly. Then, the new eigenvectors can be obtained using equation (16). Note that

[2] For simplicity we omit temporal subscript $(n+1)$ for vectors h_k and g_k.

both $R_k^{(n+1)}$ and $\Lambda_k^{(n+1)}$ are square matrices of dimension d+1, that is, we only need to solve an eigenproblem of dimension d+1 and not p. The update formulas for the variance parameters a_{kj} and b_k are then:

$$a_{kj}^{(n+1)} = \Lambda_{kj}^{(n+1)},$$
$$b_k^{(n+1)} = \frac{1}{p-d}\left(\mathrm{tr}\left(\Lambda_k^{(n+1)}\right) - \sum_{j=1}^{d} a_{kj}^{(n+1)}\right). \quad (20)$$

4. COMPARISONS WITH ONLINE EM AND CEM

We compare online MPPCA with two other online algorithms, online EM (Titterington, 1984) and online CEM (Samé, Ambroise, & Govaert, 2007). Note that these latter have not been designed to handle high-dimensional data. This benchmark was done on simulated data because we could control the real problem dimension which is not the case with real observations. An application on turbofan engine measurements is given in the next section.

For this experiment, we have generated a dataset of n = 12000 data samples in R^p based on the assumption that data live in low-dimensional subspaces, with p = 30 and K = 3. Hereafter, we refer to this dataset as D_{30}. The mixture proportions are π_1 = 0.4 and $\pi_2 = \pi_3$ = 0.3. For simplicity, we have considered that for each class, the variance is common across all dimensions, that is $a_{kj} = a_k$, for $k = 1 \ldots K$ and $j = 1 \ldots d$. We have set a_1 = 150, a_2 = 75, a_3 = 50, $b_1 = b_2 = b_3$ = 5 and μ_1 = 0, μ_2 = {0…5…0} and μ_3 = {0…−5…0}, with $\mu_1, \mu_2, \mu_3 \in R^p$. We have set the intrinsic dimension (dimension of the subspaces) at d = 2.

We also simulate a second dataset of lower dimension (p = 10) , generated with the same parameters as the former. We will refer to this new dataset as D_{10}.

Our goal was to study the behavior of the three algorithms in low dimension and then illustrate the capability of online MPPCA to cluster efficiently even in high dimension.

We have evaluated the three algorithms on the quality of their estimation of the class means and on the accuracy of the clustering produced. The quality of the estimation of the means was taken to be the square of the distance of the estimated means to the true ones, averaged over all K = 3 classes, a measure known as the Mean Square Error (MSE) in statistics

$$\mathrm{MSE}_\mu = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{p}\sum_{j=1}^{p}\left(\hat{\boldsymbol{\mu}}_{kj} - \boldsymbol{\mu}_{kj}\right)^2\right) \quad (21)$$

Online MPPCA, online EM and online CEM were initialized 30 times by a standard MPPCA, an EM and a CEM, respectively, of which the initialization giving the highest BIC value was kept.

Figure 4 and Figure 5 show the comparative performance of online MPPCA (black), online EM (red) and online CEM (blue) for the datasets D_{10} and D_{30}, respectively.

For the dataset D_{10} it is clear, both from the clustering accuracy and the MSE that online MPPCA converges faster than the other two algorithms.

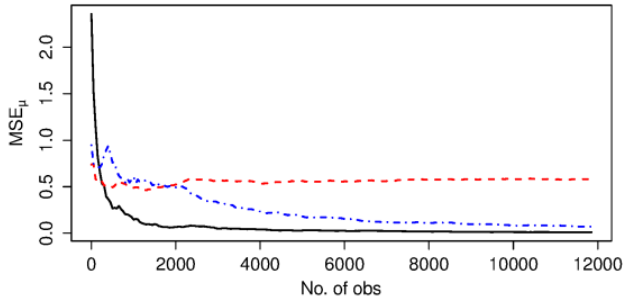


Figure 4 – Evolution of MSE for the dataset D_{10} versus the number of data samples for online MPPCA (black solid), online EM (red dashed) and online CEM (blue dotted).

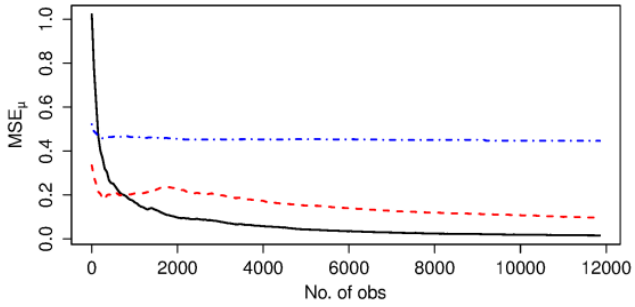


Figure 5 – Evolution of MSE for the dataset D_{30} versus the number of data samples for online MPPCA (black solid), online EM (red dashed) and online CEM (blue dotted).

5. APPLICATION TO ENGINE HEALTH MONITORING

We test the proposed method to real data issued from the aircraft engine Health Monitoring domain. The data were obtained by Snecma.

Typically, there exists different phases during a flight, called flight modes: taking-off, cruising, landing etc. Each test is actually a sequence of alternating stationary and non-stationary phases at different levels. The stationary phases correspond in general to such flight modes, while the non-stationary ones reflect the transition between two such phases. Nevertheless, a flight mode can include multiple stationary phases, that is, a stationary control on the data is not enough to detect the flight modes.

Aircraft engineers can identify these modes by looking at the data but this can be extremely time-consuming. Moreover, due to the high dimensionality of data, there can be relations that humans cannot perceive. Note that by

knowing, at any given time, in which flight mode the engine currently is, tasks like anomaly detection can be performed much more reliably, since the 'local' context of the data is also taken into account.

The experiment below (Figure 6) involves a MPPCA stage used to build a residual vector that is finally classified with a self organizing map (SOM). The score represents the distance to the corresponding class center, and the fault identification is obtained as the map cell number.

The data simulates real engine normal degradation (usual wear) to be detected by trend monitoring tools. The result appears to be pertinent for operational analysis as the MRO operator usually waits for confirmation before any customer notification.

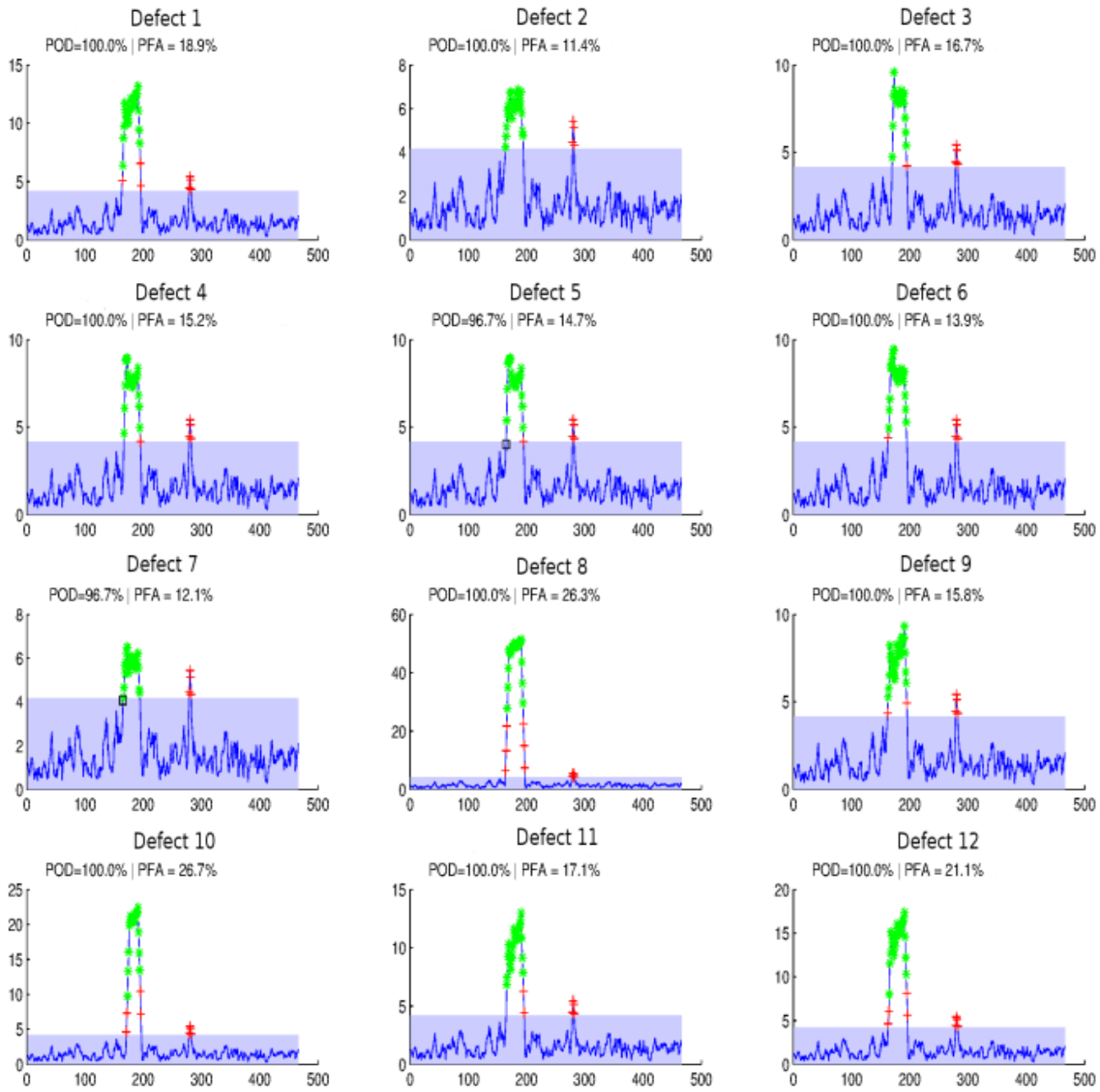


Figure 6 – Scoring and identification of trend faults using a self organizing map after MPPCA normalization. Green dots are the true detections and red ones the false alarms. POD stands for probability of detection and is given as a point to point count, as well as the PFA which is the probability of false alarm.

6. CONCLUSION

We have proposed an online inference algorithm for the MPPCA model which relies on an EM-based procedure and a probabilistic and incremental version of PCA. The proposed strategy allows to incrementally update the estimates of the MPPCA parameters at the arrival of a new data sample. It allows also providing low-dimensional visualizations of the data based on sufficient information. Model selection is also considered in the online setting through parallel computing. Numerical experiments on simulated and real data have shown that the online MPPCA algorithm performs better in high-dimensional spaces compared to existing online EM-based algorithms.

NOMENCLATURE

ACARS	Aircraft Communications Addressing and Reporting System
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
DFDR	Digital Flight Data Recorder
EM	Expectation Maximization
LASSO	Least Absolute Shrinkage and Selection Operator
MPPCA	Mixture of Probabilistic PCA
MRO	Maintenance Repair Overhaul
MSE	Mean Square Error
PCA	Principal Component Analysis
SOM	Self Organizing Map

REFERENCES

Bellas, A., Bouveyron, C., Cottrell, M., & Lacaille, J. (2013). Model-based Clustering of High-dimensional Data Streams with Online Mixture of Probabilistic PCA. *ADAC*.

Bouveyron, C., Girard, S., & Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis*, *52*(1), 502–519. doi:10.1016/j.csda.2007.02.009

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010). Aircraft engine health monitoring using Self-Organizing Maps. In *Industrial Conference on Data Mining*. Berlin.

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2011). Aircraft engine fleet monitoring using Self-Organizing Maps and Edit Distance. In *WSOM*. Espoo (Finland).

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., & Verleysen, M. (2009). Fault prediction in aircraft engines using Self- Organizing Maps. In *WSOM*. Miami (FL).

Flandrois, X., Lacaille, J., Massé, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. In *AIAA InfoTech*.

Hall, P., Marshall, A., & Martin, R. (1998). Incremental Eigenanalysis for Classification. *BMVC*, 29.1–29.10. doi:10.5244/C.12.29

Lacaille, J. (2009). Standardized failure signature for a turbofan engine. In *IEEE Aerospace conference* (p. 11/0505). Big Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J., & Côme, E. (2011). Visual Mining and Statistics for a Turbofan Engine Fleet. In *IEEE Aerospace Conference* (p. 11/0405). Big Sky (MT): IEEE.

Lacaille, J., & Gerez, V. (2011). Online Abnormality Diagnosis for real-time Implementation on Turbofan Engines and Test Cells. In *PHM*. Montreal (Canada): PHMSociety.

Lacaille, J., & Gerez, V. (2012). A Batch Detection Algorithm Installed on a Test Bench. In *PHM* (pp. 1–7). Minneapolis: PHM Society.

Samé, A., Ambroise, C., & Govaert, G. (2007). An online classification EM algorithm based on the mixture model. *Statistics and Computing*, *17*(3), 209–218. doi:10.1007/s11222-007-9017-z

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, *46*(2), 257–267.

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring. Jérôme has a PhD in Mathematics on "Neural Computation" and a HDR (habilitation à diriger des recherches) for "Algorithms Industrialization" from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial process. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.

Anastasios Bellas is a mathematic PhD form university Paris 1 Pantheon-Sorbonne. He worked on online autoadaptive clustering methodologies applied to aircraft engine measurements for Snecma. Today, Anthikos is doing is military service in Greece.

Integrated Multivariate Health Monitoring System for Helicopters Main Rotor Drives: Development and Validation with In-Service Data

Alberto Bellazzi[1], Giovanni Jacazio[2], Bruno Maino[1], Gueorgui Mihaylov[2], Franco Pellerey[3] and Massimo Sorli[2]

[1] *AgustaWestland, Cascina Costa di Samarate, VA 21017 VA , Italy*
Alberto.Bellazzi@agustawestland.com, Bruno.Maino@agustawestland.com,

[2] *Politecnico di Torino, Department of Mechanical and Aerospace Engineering, Turin, 10129 TO, Italy*
giovanni.jacazio@polito.it, gueorgui.mihaylov@polito.it, massimo.sorli@polito.it

[3] *Politecnico di Torino, Department of Mathematical Sciences, Turin, 10129 TO, Italy*
pellerey@calvino.polito.it

ABSTRACT

The implementation into service of accelerometric health monitoring systems of mechanical power drives on helicopters has shown that the generation of false failure alarms is a critical issue. The paper presents a combined application of several multivariate statistical techniques and shows how a monitoring method which integrates these tools can be successfully exploited in order to improve the reliability of the diagnostic systems. The first phase of the research activity was addressed to exploring the potential advantages of using multivariate classification/discrimination/anomaly detection methods on real world accelerometric condition monitoring data. The second phase consisted of an implementation into actual service of an innovative integrated multivariate health monitoring system.

1. INTRODUCTION

Failure diagnostics via condition monitoring on mechanical systems and components is a broad and relevant topic. Different approaches based on the development of specific sensors and data-driven methods have been applied in various contexts. For example in (K. Liu, 2013) is described the construction of a composite health index through the fusion of multiple sensor data. In many cases the calibration of reliable data-driven models is obstructed by the lack of data regarding the failure modes of the mechanical system. In such circumstances sophisticated anomaly detection and decision mechanisms might be required (see for example (Ramasso & Gouriveau, 2010)).

This project has been developed under research contract granted by AgustaWestland. It was focused on monitoring the health conditions of mechanical power drives of helicopters. Accelerometric monitoring systems have been previously installed on helicopters produced by AgustaWestland. The adopted vibration monitoring methods are based on analyzing analog signals provided by a set of accelerometers (we refer the reader to (Randall, 2011) and especially (CAA-PARER-2011, 2012)). A set of accelerometers is arranged in appropriate locations on the power drive. To each component of the power drive is associated an accelerometric analog signal. The accelerometric outputs undergo Fourier spectral decomposition and the description of the local (not global) properties of the energy distribution through the spectrum of vibrational modes leads to a set of scalar health indicators, which are supposed to detect specific damages. For example, actually physical indicators represent the energy of the spectral components corresponding to the main rotational frequency and its multiples, the energy contained in a localised energy bands etc. Other indicators, obtained from the second-level signal analysis in both time and frequency domain are related to local variations, correlations between specific spectral channels, local shape factors, signal standard deviations and signal quality.

The description of the state of each mechanical component is done by a specific set of health indicators, selected by AgustaWestland as appropriate for this scope. The health state monitoring method of each component is based on fixed critical thresholds for the values of each condition indicator. Damage alerts are generated when **any** of the indicators exceeds the threshold for certain number of measures. More in detail "yellow alert" is generated if the value of the indicator ex-

Alberto Bellazzi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ceeds certain threshold and a "red alert" is generated when the value exceeds a higher threshold. In other words the adopted monitoring method concerns a univariate (independent) **interpretation** of the health indicators.

The implementation of this health monitoring system on power drives in actual service has shown that a relatively high number of false alarms is generated, thereby requiring additional troubleshooting workload.

The purpose of this research was to develop an innovative health monitoring method **based on the same accelerometric features**, which is able to reduce to the very minimum the false positives. It is important to underline the fact that our proposal does not require installation of additional sensors in order to obtain further physical information.

The empirical observation during the employment of this monitoring/diagnostic system over a long period of time, highlighted the fact that in certain failure circumstances, groups of health indicators react simultaneously to some anomalies. For this reason, even though (by construction) the condition parameters are processed as univariate indicators, multivariate statistical techniques should be taken into account.

The efficiency of the existing diagnostic system has been improved via **third-level multivariate processing of the condition indicators**. A monitoring method which combines several multivariate statistical techniques has been developed and implemented in an efficient integrated tool. The method is able to distinguish with very high level of statistical confidence true failure situations and false anomaly alerts if these have been previously observed and diagnosed on any other aircraft of the same type.

This article provides a more detailed presentation, with addition of some later results, of the research which was preliminarly introduced in (A.Bellazzi et al., 2014).

2. IN-SERVICE DATA

The research was focused on mechanical power drives of helicopters which consist of an assembly of several gears rotating on shafts supported by ball and roller bearings.

AgustaWestland provided a **very large amount of real data** collected on 115 aircrafts of the same type flying in different conditions. The full available experimental data set consists of huge quantity of measurements of the condition indicators of several mechanical components and was collected over a period of four years and thousands of flight hours. The investigation mainly concerned the following set of power drive components in which true (confirmed by inspection of the power drive) and false alerts were detected:

- TTO Pinion, characterised by twelve relevant condition indicators. A representative calibration data set of 6291 measurements has been extracted. During the monitored period one true failure (confirmed by inspection) was observed and three false alerts were reported by the monitoring system on different helicopters.

- IGB Pin, characterised by twelve relevant condition indicators. The calibration data set is composed by 5496 measurements. Five false alerts of three different types were reported by the monitoring system.

- TGB Gear, characterised by twelve condition indicators. The data set contains 6291 measurements. During the monitored period one confirmed true failure was observed and three false alerts were reported on different aircrafts.

- TRDS, characterised by two health indicators. The calibration data set contains 3925 measurements. One confirmed damage and three false alerts has been generated.

- 2nd Stage Pin RH Brgs, characterised by six relevant condition indicators. The data set contains 6514 measurements. One true failure and two false alerts were generated.

-Oil cooler BRG, characterised by six relevant condition indicators. The calibration set is composed by 3954 measurements. The standard (univariate) control system did not report any anomalous behaviour as none of the alert thresholds has been exceeded.

- Hangar Ball Brg. characterised by nine condition indicators. The calibration set contains 4390 measurements. During the monitored period one true failure was observed and three false alerts were reported by the system.

The TRDS and the Hangar Ball Brg are monitored by the same accelerometer. The other mechanical components are monitored by different single accelerometers.

In some cases (TRDS and the Hangar Ball Brg) the individual thresholds of several health indicators were strongly exceeded (largely over the "red threshold") in a false alert situation. Unexpectedly a true damage provoked more moderate reaction of the monitoring system (values of the health indicators just above the "yellow" threshold). These cases were considered as particularly critical as the **mono-variate evaluation of the damage appears to be misleading**.

In the rest of the article the set of N health indicators of a mechanical component of a power drive will be interpreted as an element in a real N-dimensional vector space and called the **vector state** of the component.

In order to save space, the results will be illustrated by referring to some relevant examples obtained from the above components. The computations have been done by using R statistical software (for more information see (B. Everitt, 2011) and (Everitt, 2005)).

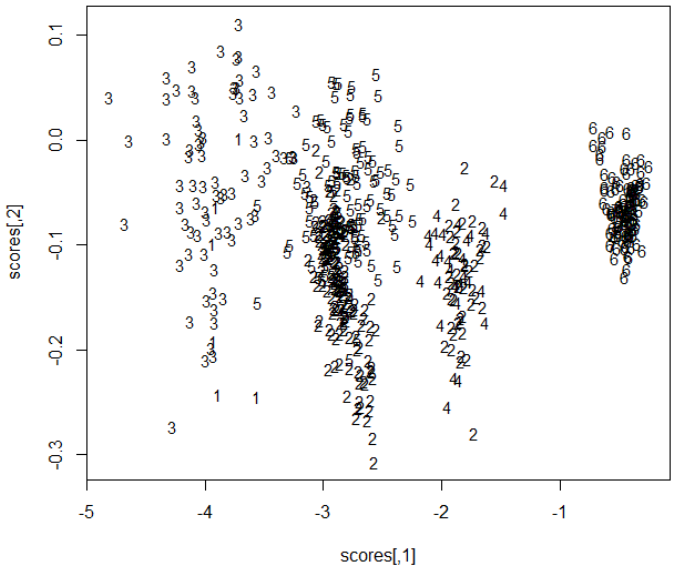


Figure 1. PCA scores of healthy operational states of TRDS component of six helicopters of the same type. Vector states measured on individual helicopters are labelled by different numbers.

3. MULTILINEAR RE-CALIBRATION AND ANOMALY DETECTION

The very first relevant problem we came across in this research program was the fact that values of the standard health indicators, which characterise the healthy operational regime of a mechanical component vary quite consistently between individual aircrafts of the same type. Typically, if compared to each-other, the vector states of the same component in healthy regime on different helicopters form neatly visible clusters inside the vector space of indicators (a striking illustration is given on Fig. 1). Observe that, in this specific case, the individual helicopter clusters spread along the direction determined by the first principal component. This means that by far the most consistent portion of the variance in the data set of healthy operational states can be attributed to differences between individual aircrafts.

The fact that healthy operational states of a power drive installed on different aircrafts cannot be compared, makes **impossible the calibration of any sort of statistical model, based on historical collection of vector states** measured on a fleet of helicopters. Moreover the mechanical components selected for the investigation are typically subject to a very low number of failures. A calibration and a validation of a reliable multivariate model on **each single aircraft** appears therefore as extremely unrealistic.

A solution to these problems is described herein.

Besides the set of component vectors, a historical collection of simultaneous measurements of the following parameters of the operational condition of each aircraft was available: Engine 1 Torque, Engine 2 Torque, Rotor Speed, Roll Angle, Pitch Angle, True Airspeed, Radio Altitude, Vertical Speed, Normal Acceleration, Density Altitude, Tail Rotor Torque, Main Rotor Torque, Roll Rate, Pitch Rate, Yaw Rate, Longitudinal Acceleration.

It has been hypothesised that the accelerometric measurements are in some extent influenced by the environmental state of the aircraft. In order to test that hypothesis, **canonical correlation analysis** has been applied on the available data set.

The canonical correlation method describes the interconnection between two random vector variables by means of a double set of latent variables (directions in the corresponding state vector spaces). Those latent variables reproduce the structure of the correlations between the "physical" observed variables of different groups, minimising in the meanwhile the impact of the correlations between variables in the same group. These latent variables are called canonical components and are ordered according to the magnitude of the common eigenvalues of certain matrices, which has been defined by Hotelling in (Hotelling, 1936). The observable parametrisation of the physical vector states of the variables in the groups can be replaced by a more synthetic one, which is obtained in terms of projections in the directions determined by the canonical components. The linear correlations established between the latent variables, constructed in such a way, are called **canonical correlations** of the model. As an example, the list the canonical correlations obtained by analysing the interconnections between the environmental state vectors and the vector states a TGB gear is displayed below (values of the canonical correlation coefficients close to 0 indicate low correlation, values close to 1 indicate high correlation between canonical variables):

$$\begin{array}{ll} \rho(a_1 b_1) = 0,99999838 & \rho(a_2 b_2) = 0,74719544 \\ \rho(a_3 b_3) = 0,60608554 & \rho(a_4 b_4) = 0,47571818 \\ \rho(a_5 b_5) = 0,39483775 & \rho(a_6 b_6) = 0,37293685 \\ \rho(a_7 b_7) = 0,26062950 & \rho(a_8 b_8) = 0,15779505 \\ \rho(a_9 b_9) = 0,13292464 & \rho(a_{10} b_{10}) = 0,10704979 \\ \rho(a_{11} b_{11}) = 0,06135586 & \rho(a_{12} b_{12}) = 0,02884099 \end{array}$$

In each of the analysed cases the first canonical correlation is extremely high. This fact, considered the high number of dimensions, can be considered as accidental. More relevantly it has been observed that many components are characterised by three or four canonical correlations with considerably high values (over 0,5). This fact is much more meaningful with respect to the interrelations between the environmental vector state and the component vector state. Unlikely, in some cases (Hangar Ball Brg) the canonical correlation profile is characterised by very low second canonical correlation.

The existence of relevant multi-correlation between the aircraft states and component states led us to the construction

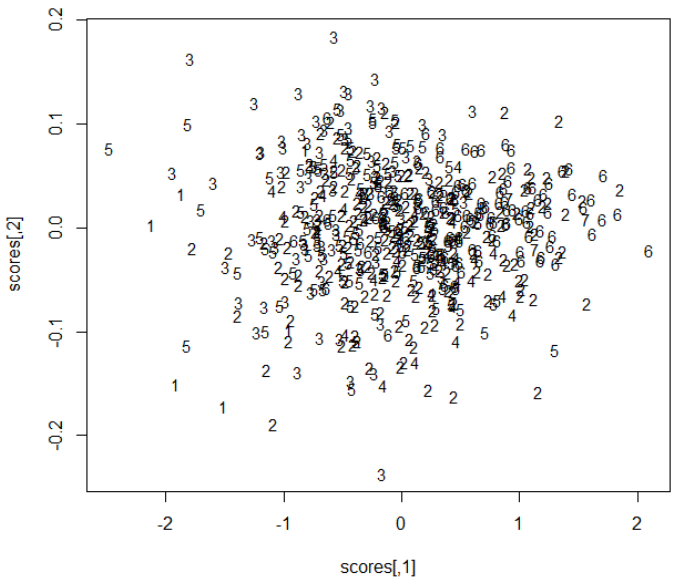


Figure 2. PCA scores of healthy operational states of TRDS of the same six helicopters after linear re-calibration. Again vector states measured on individual helicopters are labelled by different numbers.

of what has been called a **multilinear filter**. A liner map $f : R^{17} \longrightarrow R^N$ (where N is the dimension of the component vector) which provides a "predicted" component vector state in correspondence to each environmental state has been calibrated. The k-th row of the matrix associated to this linear map (with respect to the canonical basis of physical variables of the state vector space) represents the coefficients of a multiple liner regression of the k-th component of a state vector over the set of environmental parameters. The calibration is done in healthy conditions and the analysis is then performed in terms of residuals with respect to the predicted value.

If the reader compares Fig. 1 to Fig. 2, will observe that as a consequence of re-calibration, scores of healthy operational states measured on different helicopters slightly concentrate (compare the scales of the diagrams) and mix together quite uniformly. Similar effects are observed for all the mechanical components, for which the canonical correlation analysis reveals considerable level of linear correlation. Linear re-calibration makes vector states measured on individual helicopters of the same type **comparable**. A specific situation on an aircraft can be compared to analogous situation on another aircraft.

One of the standard anomaly detection tools in multivariate statistics, based on the statistically relevant Mahalanobis distance, is the so called multidimensional Shewhart control chart (we refer the reader to (Shewhart, 1931) and (Shewhart, 1986)). Control charts are based on an evaluation of the likelihood on a single event in the context of a random process. Consider a vector space endowed with a probability distribution f and a sample (a process) of random vectors $(X)_i \in V$. As long as the sample vectors belong to regions where the probability density is judged sufficiently high, the process is considered **under control**, or **out of control** otherwise. Under certain symmetry assumptions on the probability distribution density f, control charts can be implemented as distance based statistical methods. A state X is considered out of control if it is "far enough" from the expectation value of the distribution of the ordinary regime of the process.

In a population characterised by a **multidimensional Gauss distribution**, the Mahalanobis distances from the mean value follow the $T_k^2(n)$ distribution. Moreover there is an exact correspondence between the $T_k^2(n)$ distribution and the Snedekor-Fisher variable F:

$$\frac{n-k+1}{nk}\, T_k^2(n) \cong F_{k,n-k+1},$$

which is exploited for inference purposes. This means that plausibility of a state is compared to a statistical significance level imposed on the values of the $F_{k,n-k+1}$ distribution. Distances which exceed the one corresponding to the significance level indicate a phenomenon which is very improbable under the hypothesis of being a manifestation of the ordinary regime of the process. For this reason such a state is judged as a modification of the process due to not accidental causes.

The normality of the distribution of the healthy states of the mechanical component is a necessary condition for the application of a Shewhart control chart. On Fig. 3 are displayed the scores of the unfiltered healthy operational states of a TGB gear of one helicopter with respect to the first two principal components. The reader can observe that the cluster of PCA scores is characterised by an asymmetric "tail" in the direction determined by the second principal component. The PCA scores of the healthy states of the same component after linear re-calibration procedure are displayed on Fig. 4. The first obvious consequence of linear filtering is that the shape of the cluster of PCA score components becomes more ellipsoidal (recall that level sets of the Gaussian distribution are ellipsoids).

The extent to which the filtered healthy operational states of each component of the power drive fit with a multidimensional Gauss distribution has been tested. This fact was verified by various multivariate normality tests like Kolmogorov-Smirnoff, Jarque-Bera etc. (see (Kolmogorov, 1936; A. Justel, 1997; C. M. Jarque, 1987)). It has been observed that the distribution of filtered healthy operational states of a component of a single helicopter can be considered as Normal with very high level of statistical confidence (p-value around 2×10^{-15}). Analogous behaviour was observed in all the analysed mechanical components.

The above results can be interpreted by saying that the linear re-calibration procedure **filters the deterministic impact**

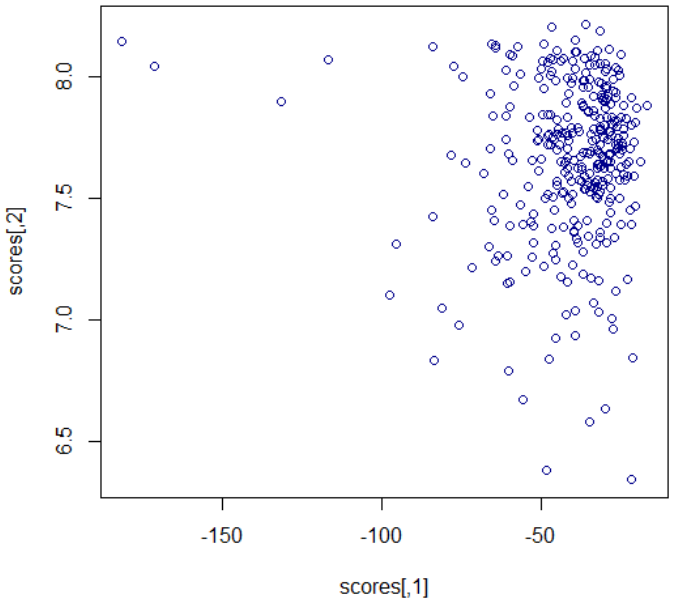


Figure 3. PCA scores of healthy operational states of the TGB gear of a single helicopter before linear re-calibration.

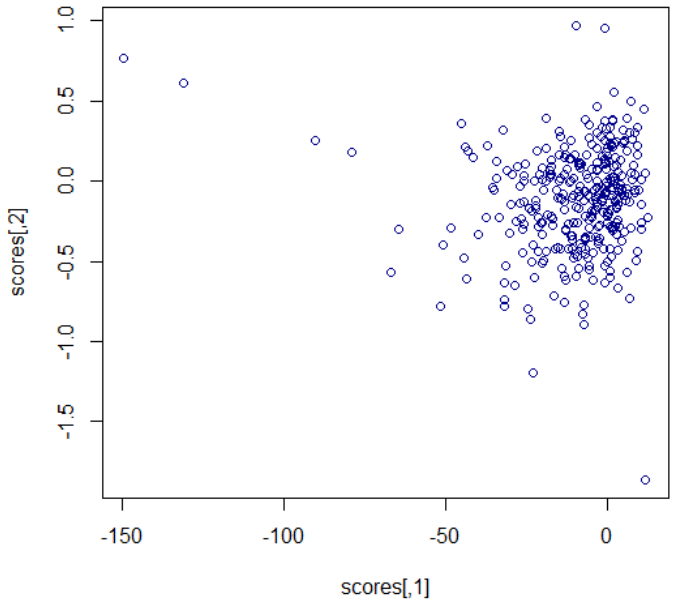


Figure 4. PCA scores of healthy operational states of the TGB gear of a single helicopter after linear re-calibration.

of the general state of the aircraft onto the accelerometric measurements. Once filtered the influence of the specific exploiting regime of the aircraft, the intrinsic variability of the healthy operational states of each mechanical component can be modelled over a random (white) noise process.

Fig. 2 illustrates the fact that analogous remark regards the set of filtered healthy operational states of the same component installed on different helicopters of the same type. They are normally distributed with roughly the same statistical confidence but with slightly higher variability.

Shewhart control charts have been calibrated on the set of healthy operational states of each mechanical component on a single helicopter. A small portion (less than 2%) of healthy vector states exceed the control limit. The same control chart was applied to healthy operational states of the same power drive, installed on other "twin" helicopters and bigger portion of states was judged out of control (15% for the Hangar Ball Brg). This means that even though linearly filtered data are used, there are still residual differences between the healthy regimes of components of different aircrafts. The same control chart has been also validated in the context of anomalous situations occurred on the same helicopter with very good results. In the case of Hangar Ball Brg roughly 73% of the anomalous states were judged out of control.

In conclusion, anomaly detection method based on a Shewhart control chart must be calibrated on each single helicopter. A software tool implementing a multivariate **self-learning** Shewhart control chart, which calibrates itself automatically on the healthy regime of a single mechanical component and highlights anomalous states, has been produced. The program computes automatically the upper control limit by means of a Gaussian approximation of the Fisher-Snedecor distribution.

In many cases (especially TRDS and Hangar Ball Brg) the Mahalanobis distance between states corresponding to false alerts and the mean value of the healthy regime exceeds the distance of the true damage states. For this reason the multivariate self-learning Shewhart control chart is an excellent tool for the detection of anomalous situations, but it is not sufficient for the discrimination of true failure states and anomaly alerts which do not correspond to a failure. Thus, additional discrimination statistical tools have been applied, as described later on. In the following sections of this article statistical models are calibrated and validated on filtered data.

The re-calibration filter can be made even more powerful by applying higher order regression of the health indicators over the set of environmental parameters of the aircraft. As an example, the reader can compare the previous canonical correlations of the linear filter of the TGB gear with the following canonical correlations of a quadratic multiple regression on the same component:

$$\begin{array}{ll} \rho(a_1 b_1) = 0,9999989 & \rho(a_2 b_2) = 0,8211778 \\ \rho(a_3 b_3) = 0,7046168 & \rho(a_4 b_4) = 0,6313677 \\ \rho(a_5 b_5) = 0,5325972 & \rho(a_6 b_6) = 0,4903959 \\ \rho(a_7 b_7) = 0,4192670 & \rho(a_8 b_8) = 0,4000518 \\ \rho(a_9 b_9) = 0,3749749 & \rho(a_{10} b_{10}) = 0,3463869 \\ \rho(a_{11} b_{11}) = 0,2774180 & \rho(a_{12} b_{12}) = 0,2452053 \end{array}$$

In conclusion the quite encouraging results obtained by linear re-calibration procedure can be even further improved.

4. MULTIVARIATE DISCRIMINATION METHODS

The linear re-calibration strongly reduces the differences between the healthy operational regime of power drives installed on aircrafts of the same type. This fact enables us to apply a set of standard multivariate statistical methods on a historical database of a fleet of helicopters. For a detailed description of those techniques the reader can refer to the following texts (Ferrell, 1979; Rencher, 2002; Timm, 2002; W. K. Härdle, 2012; Izenman, 2008).

In this study a particular geometric viewpoint on multivariate statistics has been adopted, as long as an Euclidean approach (or a more general metric geometry) provides some very useful intuitions on multivariate methods (see (Wickens, 1995) and (Epps, 1993)). We also refer the reader to (Tyurin, 2009), where a more intrinsic (coordinate free) geometric prospective on multivariate statistics is presented. In this context the analysis has been developed in terms of directions (random variables) and projections (magnitudes) onto relevant subspaces of the space of state vectors. Analogously but independently on the work exposed in (Gniazdowski, 2013) our approach interprets correlations as angles, but further radicalises this viewpoint by identifying statistical variables (both observable and latent-ones) in terms of real projective classes in a space of random vectors.

4.1. Structure of variance

The complete set of available states (healthy, true failures, false alerts) of each mechanical component was processed by Principal Component Analysis (PCA). This method is a direct implication of the Spectral Theorem in linear algebra. Principal components are the directions in the vector space of random variables, which maximise the variability of the data set. This technique highlights existing **spontaneous clusterings** in the variance structure of the data set. On Fig. 5 is displayed an example of scores of complete data sets on the subspace generated by the first three principal components.

A remarkable fact is that, after filtering, healthy operational states measured of many helicopters of the same type form a well-defined (green) cluster (see Fig. 5). Furthermore, there is an evident spontaneous clustering of the healthy and the anomalous true/false anomalous states. PCA leads to a consistent dimensional reduction in the space of states. Equations of linear and quadratic separation surfaces between the projections of the group clusters have been easily worked out and simple control methods have been based on the spontaneous clustering for each of the analysed mechanical components.

In the "critical case" of Hangar Ball Brg the projections on the subspace generated by the first and the second principal component do not reveal a significant clustering of the vector states. Nevertheless there is a relevant spontaneous clustering of the scores with respect to the second and the third principal

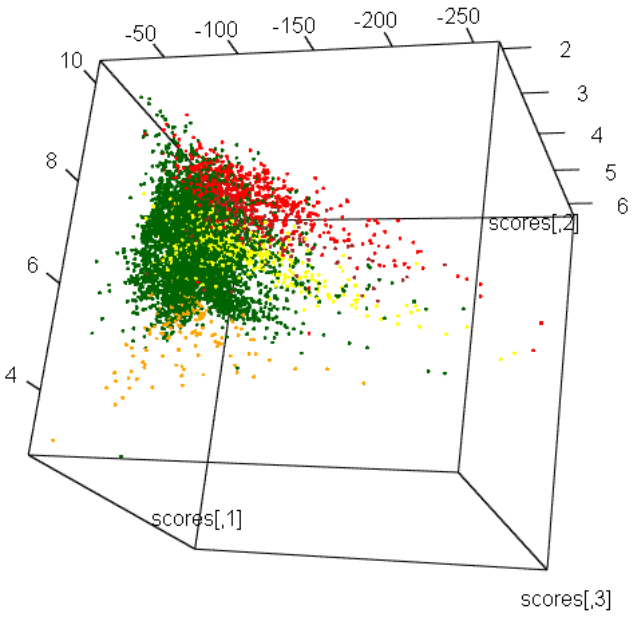


Figure 5. PCA scores of the states of a TGB Gear. Green dots represent scores of healthy operational states measured on 18 helicopters, red dots - true failure states measured on one of those helicopters, yellow dots - false alert on one of those helicopters.

components which was exploited in order to define discrimination conditions (see Fig. 6).

On Fig. 7 are displayed PCA scores of a 2nd Stage Pin RH Brgs measured on a number of twin helicopters. The ordinary healthy operational states arrange in a very compact cluster. The set of blue dots represents a false alert occurred on one helicopter of the fleet. The yellow and the red dots represent anomalous states of the component measured on another helicopter of the fleet. In this case the chronological analysis of the data set led us to the following interpretation. An early fault (cluster of yellow dots) evolves towards a failure (cluster of red dots). The distinction between false and true anomalies is extremely sharp in this case and the direction in which the projections of true anomalous states spread in the space generated by the first three principal components is **indicative regarding the type of failure even before the definitive failure occurs**.

The structure of variance in the data sets has been further explored by applying multivariate discrimination methods like Liner Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) (see (W. K. Härdle, 2012)). The standard Fisher's linear discriminant model is based on a linear transformation of the vector space, which maximises the differences between the transformed sample mean values of the distinguished groups. In other words LDA defines a new basis (a set of latent variables) such that the impact of the **between** component of the covariance matrix gets maximised at the

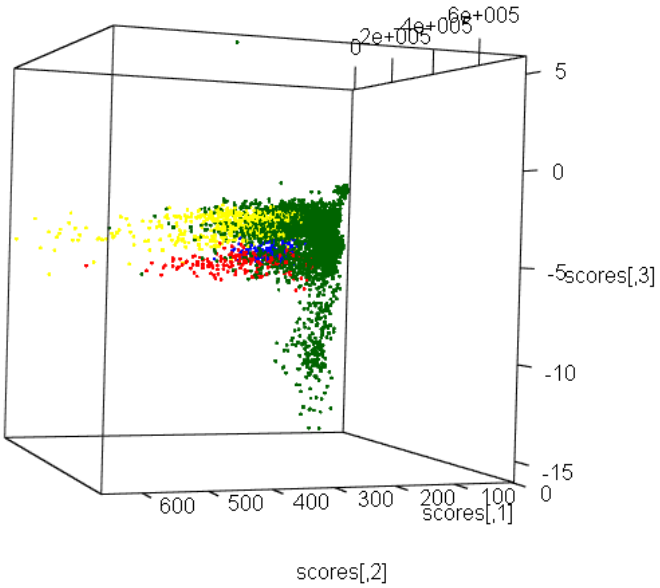


Figure 6. PCA scores of the states of a Hangar Ball Brg. Green dots are scores of healthy operational states from 18 helicopters, red dots - true failure states of one of those helicopters, yellow and blue dots are different false alerts on two helicopters.

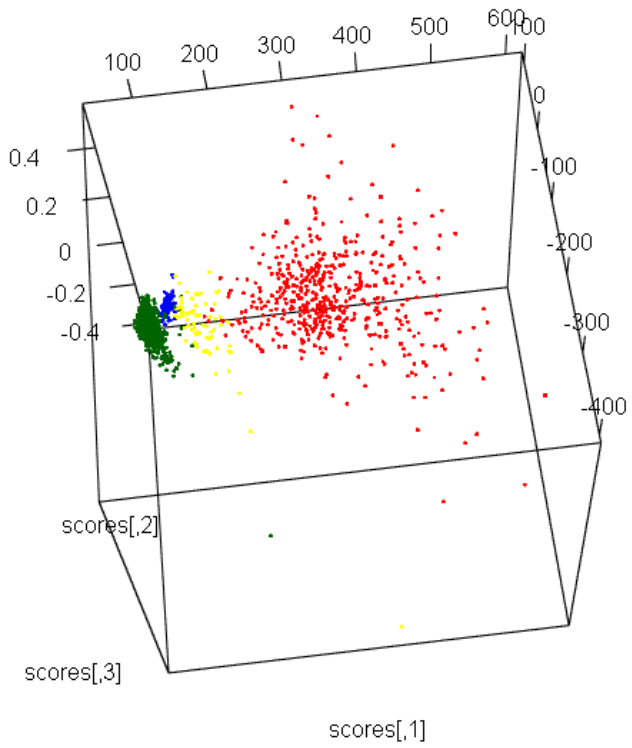


Figure 7. A 2nd Stage Pin RH Brgs fault and failure detection by means of PCA. Green dots represent healthy states measured on a fleet of helicopters, red dots - true failure occurred on one of those helicopters, yellow dots - fault states, blue dots - false alert occurred on one helicopter of the fleet.

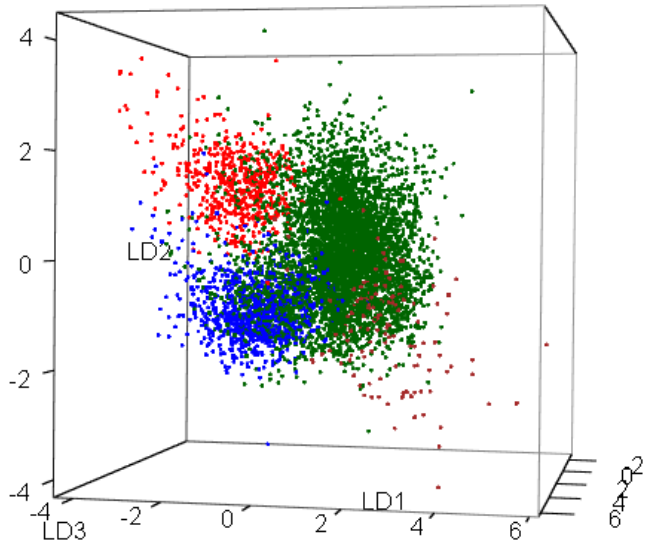


Figure 8. LDA scores of TGB Gear. Green dots are scores of healthy operational states measured on 18 helicopters, red dots - true failure states measured on one of those helicopters, blue dots - false alert on one of those helicopters.

Table 1. Leave-one-out LDA re-classification of 2nd Stage Pin RH Brg vector states

real \ classified as	false alert	healthy	true failure
false alert	74	0	0
healthy	1	1869	0
true failure	8	67	495

expense of the **within** component. The decision boundaries of LDA are linear affine subvarieties of the space of states.

The set of component state vectors has been divided into three groups, healthy operational states, false alerts and true failures. On Fig. 8 are displayed projections of TGB Gear states onto the subspace generated by the first three linear discriminant functions.

The calibrated linear discriminant models were validated by standard leave-one-out procedure using the complete data set of the fleet. On Table 1, and Table 2 are displayed some examples of LDA re-classification results.

There is a well-known quadratic classifier which exploits the minimisation of the Mahalanobis distance (with some corrections) from the mean vectors of the pre-assigned groups (see (Rencher, 2002)). In general QDA is a more flexible and precise method than LDA. Its decision boundaries are determined by the equality condition (equal probability) of the quadratic discriminant functions and are therefore (portions of) quadric hyper-surfaces in the space of states, typically ellipsoids or paraboloids. On Table 3 and Table 5 ere displayed some examples leave-one-out quadratic discriminant validation results.

Table 2. Leave-one-out LDA re-classification of Hangar Ball Brg vector states

real \ classified as	false alert	healthy	true failure
false alert	54	6	4
healthy operat.	29	1513	20
true damage	5	49	117

Table 3. Leave-one-out QDA re-classification of 2nd Stage Pin RH Brg vector states

real \ classified as	false alert	healthy	true failure
false alert	74	0	0
healthy	0	1860	10
true failure	0	0	570

Table 4. Leave-one-out QDA re-classification of Hangar Ball Brg vector states

real \ classified as	false alert	healthy	true failure
false alert	60	2	2
healthy operat.	63	1430	69
true damage	2	33	136

Table 5. Leave-one-out QDA re-classification, fault and failure detection of a 2nd Stage Pin RH Brgs

real \ classified as	normal	false alert	fault	true failure
normal	1860	0	10	0
false alert	0	74	0	0
fault	0	0	75	0
true failure	0	0	0	495

The results obtained by both LDA and QDA leave-one-out cross validation are quite encouraging, especially because of the small portion of miss-classified true failure states. In the "critical" case of the Hangar Ball Brg both methods provide statistically significative number of correctly classified true failure states. This means that **true failure can be unambiguously detected**.

Several validation procedures based on splitting of the huge initial data set into calibration and validation data subsets have been applied in order to compare different helicopters of the same type. The results provided by the alternative validation methods are basically analogous to the leave-one-out and are therefore quite satisfying.

Both discrimination methods provide excellent results in the case of the fault and failure detection of the 2nd Stage Pin RH Brgs. The component states were divided into four groups (healthy/false alert/true fault/true failure) and the results of a QDA re-classification is displayed on Table 5.

4.2. Failure detection via canonical correlation

Canonical correlation analysis can be employed for detecting anomalies. Suppose that the healthy operative regime of a process is characterised by a strong correlation between vector variables X and Y. In such case one estimates the values of Y starting from known values of X by a suitable linear model. If Y assumes "unexpected" values i.e. its behaviour contrasts with the established correlation, this fact can be considered as a manifestation of some anomaly.

The reader can notice the analogy with the so called **consistency based anomaly detection** methods in which the deviations or inconsistences with a fixed functional model are considered as anomalies. In this study a multilinear model, which returns a state of a mechanical component as a function of the environmental parameters of the helicopter has been calibrated. The hypothesis that anomalous behaviour of a mechanical component is uncorrelated with the environmental data, i.e. is a manifestation of an inconsistency with the linear model was then tested. One would expect that the linear correlations between the environmental parameters and the components health indicators should decrease in presence of anomalous behaviour of the component. Therefore the data sets of healthy states and data sets containing anomalous states have been compared in order to establish whether the relevant (high) linear correlation coefficients decrease.

The situation which emerges from this procedure appears a bit chaotic. For the TRDS the linear correlation is very strong and the values of the coefficients drastically drop in mixed regime which contains true failure states. For the IGB pin the linear correlation is strong, the correlations in mixed regime get certainly worse, but monitoring of that component did not give evidence for real failures, so the measured anomalies correspond to false alerts. The TGB gear is characterised by relatively high values of the significant correlation coefficients and its mixed regime contains a true failure, but it seems that the second canonical correlation slightly improves in mixed regime.

In conclusion, for components for which the linear correlation with the environmental states is particularly high, the theoretical hypothesis is confirmed. This means that for those components the canonical correlation method can be considered as a supplementary anomaly detection resource. We expect that higher order filtering models as the one previously mentioned, will provide more unambiguous results.

4.3. Structure of covariance

In this study, a particular modification of the covariance matrix of the vector states of some mechanical components in case of anomalous measurements has been observed. The states of true damage are often characterised by increased correlation of certain vector components. The behaviour of the correlation matrix appeared slightly different in the case of false anomaly reports.

A possible explanation of this phenomenon could be given

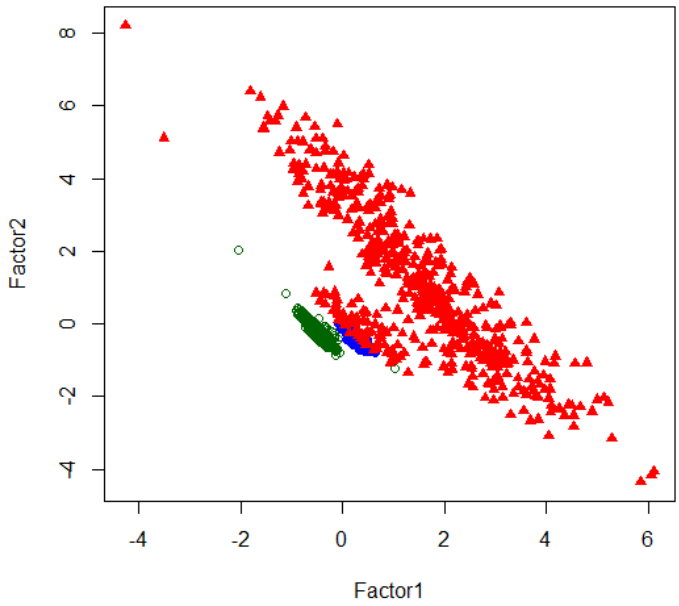


Figure 9. Bartlett factor scores of the 2nd Stage Pin RH Brgs. Green circles represent scores of healthy operational states measured on several helicopters, red triangles - true failure states measured on one of those helicopters, blue dots - false alert on one of those helicopters.

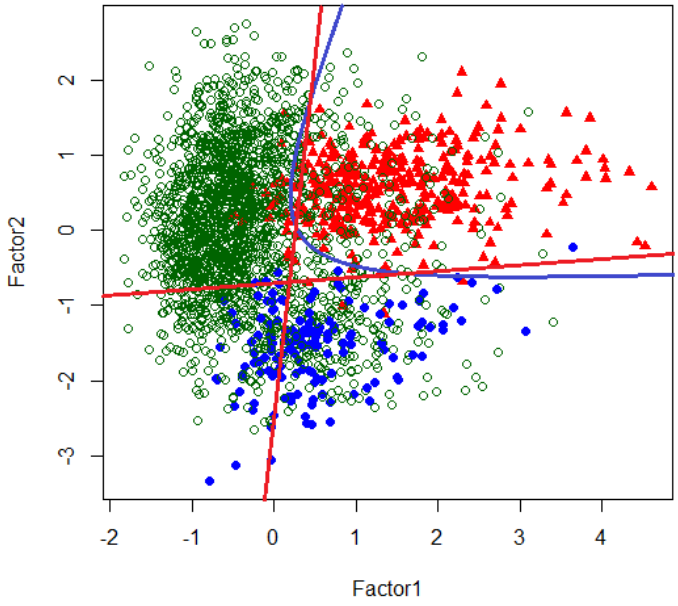


Figure 10. Bartlett factor scores of TGB Gear. Green circles represent scores of healthy operational states measured on 18 helicopters, red triangles - true failure states measured on one of those helicopters, blue dots - false alert on one of those helicopters.

if, in the case of true failure, different health indicators react simultaneously in a consistent and correlated way (failure states provoke an enhancement of certain elements of the correlation matrix). On the contrary false alerts can be interpreted as anomalous measurements not necessarily induced by a consistent reaction of the monitoring system.

The main purpose of the so called *factor analysis* consists of describing the structure of the correlations of a set of random variables by means of a small number of underlying uncorrelated latent variables called **factors**. In such sense it is analogous to the methods of principal component analysis, in which the structure of the variance in the sample is described by dimensional reduction. In this case the aim is obtaining a significant description of the structure of the covariance in the multivariate statistical sample in suitable subspace.

A compact multidimensional version of the defining equation of a factor model is:

$$X = \mu + \Lambda F + U,$$

where X denotes a k-dimensional vector random variable, Λ is a $k \times m$ matrix and U is the vector of specific factors. The matrix Λ is called the loadings matrix of the model.

The columns of Λ have an immediate geometric interpretation, they represent vectors which detect the directions of the latent factor multivariate variables. The vector variable F is nothing else but the k-uple of the projections (components) of the physical vector state X along those directions. In other words X is decomposed in certain relevant directions and its projections represent magnitudes of new variables. The vector F can be itself considered as a random m-dimensional vector variable.

The above expression only apparently resembles a multivariate linear model, in fact care must be taken as the whole expression in the second term of this equation is based on latent i.e. unobservable variables.

In this case standard recursive methods for the calibration of factor models have been applied and canonical factor models have been defined on the set of state vectors. Typically the calibration of factor model based on two factors was possible (the calibration procedure converges), but in some cases as the one of the Hangar Ball Brg, the iterative procedure does not converge with two but with three factors.

In terms of projections onto the space generated by the principal factors, the theoretical hypothesis translates in the following way. One could expect that the projections of the healthy operational cluster (near by the origin) and true failure cluster (away from the origin) onto the subspace generated by the principal factors show different characteristic profiles. The direction in which failure states projections spread away from the origin is indicative regarding the correlation modifications introduced by the simultaneous reaction to a damage. The shape of the cluster of healthy operational states characterises the intrinsic covariance structure of the component. In this context we expect that anomalous or false alerts should re-

veal some sort of irregular behaviour.

On Fig. 9 and Fig. 10 are shown the projections of the states of the 2nd Stage Pin RH Brgs and the TGB gear. Relevant clustering is rather visible in both cases. Projections (factor scores) of true failure states spread away from the origin in a direction, which is characteristic for the modified covariance structure. The investigation based on in-service data substantially confirmed the theoretical hypothesis. It is easy to work out linear or quadratic decision boundaries on factor scores as those displayed on Fig. 10.

In the case of Hangar Ball Brg the factor scores of the healthy operational states concentrate again near by the origin and the anomalous states spread far from it. Nevertheless these projections do not reveal a striking separation between true and false alert states.

In conclusion, for some mechanical components the covariance structure of the vector data set provides further useful resources for defining discriminant procedures.

5. PROJECTIVE STRUCTURE OF DATA SETS

Random variables has been interpreted as real projective classes in a vector space. From this viewpoint it is natural to hypothesise that the correlation structure of the data set can be better understood in terms of directions of the state vectors. In this context the module of a vector state plays a minor role and a direction in a vector space can be identified by a unit vector. In order to test this hypothesis, an original "experiment" has been performed. Normalised state vectors has been considered, the set of N-dimensional vector states arranges over an $(N-1)$-dimensional sphere and factor models on the set of unit vector states have been calibrated.

An obvious effect of the spherical re-definition is a sort of compactification of the operational state clusters (Fig. 11). The hypothesis on the characteristic variations of the covariance structure appears rather plausible. In fact points representing healthy operational states and true damage situations form well-defined compact clusters.

Remarkably, as a consequence of this original procedure, the discrimination between true and false alerts becomes much more striking (compare Fig. 11 to Fig. 9). In this new situation the definition of the linear discriminant conditions appears even easier and more precise with respect to the previous factor models.

The typical behaviour of the unit states of a power drive component is that true damage states condense in a compact region inside the scatter-plot cluster of states. It is often easy to work-out a discriminant condition based on the affinity to that specific compact region. On Fig. 12 is shown the case of a TGB Gear.

Another considerable advantage of the normalisation of the

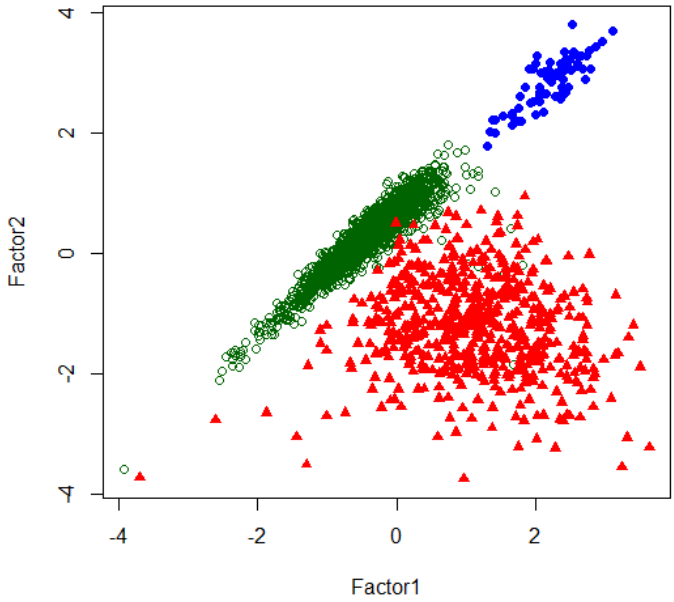


Figure 11. Bartlett type scores of unit states of a 2nd Stage Pin RH Brgs. Green circles are scores of healthy operational states measured on several helicopters, red triangles - true failure states measured on one of those helicopters and blue dots - false alert on one of those helicopters.

Table 6. Leave-one-out QDA re-classification of Hangar Ball Brg unit vector states

real \ classified as	false alert	healthy	true failure
false alert	60	2	2
healthy operat.	63	1432	67
true damage	2	33	136

vector states is the elimination of the large spreading of false anomalous alerts far from the mean value of the healthy operational regime. In this context LDA leads to precisely the same classification results, but remarkably QDA of the unit vector states of the "critical case" Hangar Ball Brg produces a slight improvement (compare Table 6 to Table 5).

In conclusion, this peculiar mathematical experiment led to interesting and in some cases unexpected, potentially useful results. The principal factor analysis on unit states gives further, often relevant, information on the anomalous behaviour of some mechanical components, and can be therefore integrated in a control procedure.

6. IMPLEMENTATION

The statistical techniques tested over the available vector data set are based on different mathematical constructions. They provide different and therefore not overabundant results. For this reason the above techniques have been combined in a software implementation of an integrated control process in the following way:

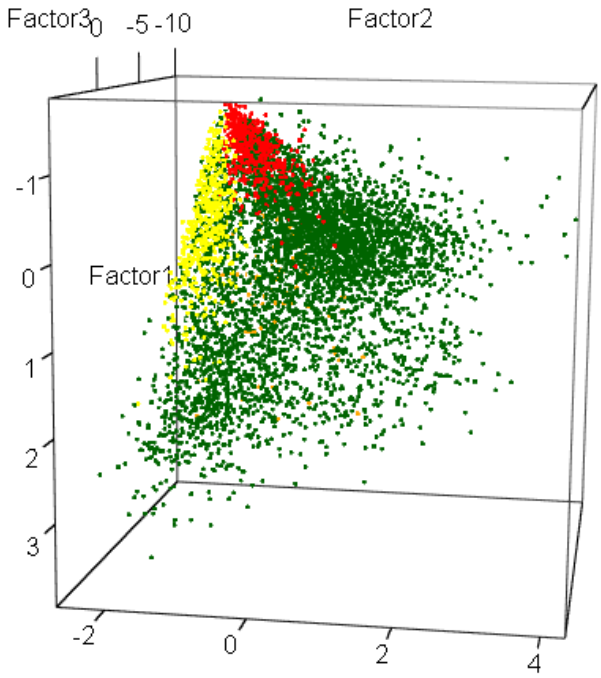


Figure 12. Bartlett type scores of unit states of a TGB Gear. Green dots represent scores of healthy operational states measured on 18 helicopters, red dots - true failure states measured on one of those helicopters, blue dots - false alert on one of those helicopters.

1. Anomaly detection by means of a self-learning Shewhart control chart. A problem highlighted by the experts of AgustaWestland consists of the fact that the healthy operational regime of some power drives on certain helicopters is characterised by very high values of the health indicators. Such values would be considered as anomalous if compared to other helicopters or to some a priori fixed threshold values. This ambiguity is completely removed by the self learning individual calibration of the control chart. Any vector state judged in control contributes to the real time re-calibration of the control chart i.e. the control chart keeps learning.

2. Anomaly classification based on discriminant methods calibrated and validated over the entire fleet. A vector state judged as anomalous undergoes evaluation based on a set of different discriminant techniques which can regard both the variance and the covariance structure of the calibration data sets (PCA, LDA, QDA, factor scores). A state classified as false alert does not generate an alert.

3. Evaluation. For different power drives, different discriminant methods appear as more efficient. An integrated parallel application of all the calibrated discriminant method is more powerful discrimination tool than the individual application of any single technique. A pre-alert status is produced by a suitable combination of discriminant outputs. Such a combination is chosen in order to maximise the efficiency of the integrated control system.

The integrated control process was tested on a series of real cases contained in the historical database of AgustaWestland. In the cases of the TGB gear and 2nd Stage Pin RH Brgs the integrated discriminant method judges a state as true failure i.e. generates a pre-alert if **each** discriminant method classifies it as a true failure. With this requirement only 3% of the measured states were miss-classified. In the most difficult case of Hangar Ball Brg a pre-alert is produced in 13% of the healthy states, in 28% of the previous false alerts and in 65% of the true failure states. The current univariate version of the control system generates an alert if the values of the health indicators exceed the alarm thresholds in a fixed proportion (usually 2/3) in a number of consecutive measurements. In the integrated method **the density of true failure outputs** required for a failure alarm can be rigorously deduced directly from these last overall validation results. For example, in the case of Hangar Ball Brg 1/2 appears as a suitable proportion.

An engineering software tool, which implements both the control process and the calibration of the parameters of the control routine for each component of the monitored power drives, has been produced.

7. CONCLUSIONS AND FUTURE DEVELOPMENT

The study has highlighted the advantages of this third-level multivariate approach. An efficient control process is based on an integration of several classification techniques. Even in those cases in which true failures and false alerts show misleading univariate profiles, multivariate techniques are able to distinguish them with **very high level** of statistical confidence.

In view of the results obtained by this research, an integrated multivariate health monitoring system is **currently in phase of implementation into actual service** on two models of helicopters produced by AgustaWestland.

The **elimination of the deterministic influence of the environmental states** of the helicopter determines two huge advantages:

1. After filtering the individual behaviour of each power drive can be very faithfully modelled over a random noise process. **The a priori threshold-based anomaly detection was therefore completely replaced by self-learning Shewhart control charts which operate individually on each mechanical component on each aircraft**.

2. Filtering gives the possibility to compare rigorously vector states measured on individual helicopters in different flight conditions. Once guaranteed the homogeneity of the measured data, **powerful classification and discriminant models can be calibrated on historical data obtained from many helicopters**. These models are applied in the context of a

control and diagnostics process over the entire fleet of helicopters. When relevant new data are collected, the statistical models should be updated and improved by re-calibration on a larger and more detailed data set. **Once a precise anomaly gets observed and diagnosed on one aircraft of the fleet, it can be diagnosed elsewhere** by means of its specific multivariate health condition profile.

The analysis of the results of this research from the viewpoint of the a posteriori **prognostics and health monitoring validation** of a diagnostic system will be a very interesting task. This is an extremely relevant topic which concerns the evaluation of the efficiency of the constructed health indicators i.e. how exhaustively they describe the state of the mechanical component (an observability problem). The investigation shows that the univariate processing of the health indicators could provoke a loss of relevant information. The results of this work show that besides the obvious advantages of direct multivariate processing, there is an interesting possibility to define and apply multivariate health monitoring validation protocols which aim to improve the efficiency of each individual health indicator by minimising the overall loss of information.

Another possibility, quite worthy to be explored in future, consists of by-passing the phase of construction of specific condition indicators by adopting a completely **multivariate spectroscopic approach** to the processing of the accelerometric signals. The calibration/monitoring software engineering tool which has been built in the context of this work can be directly applied without any modification in the context of such an alternative approach.

REFERENCES

A.Bellazzi, et al. (2014). A multivariate statistical approach to the implementation of a health monitoring system of mechanical power drives. In *Proceedings of the European Conference of the prognostics and Health Management Society ISBN-978-1-936263-16-5.*

A. Justel, R. Z., D. Pena. (1997). A multivariate Kolmogorov-Smirnov test of goodness of fit. *Statistics and Probability Letters*, *35*(3), 251259.

B. Everitt, T. H. (2011). *An introduction to applied multivariate analysis with R*. Springer.

CAA-PARER-2011. (2012). *Intelligent management of helicopter vibration health monitoring data* (Vol. 01 Based on a report prepared for the CAA by GE Aviation Systems Limited; Tech. Rep. No. ISBN 978 0 11792 403 1). Civil Aviation Authority - Safety Regulation Group.

C. M. Jarque, A. K. B. (1987). A test for normality of observations and regression residuals. *International Statistical Review*, *55*(2), 163172.

Epps, T. W. (1993). Characteristic functions and their empirical counterparts: Geometrical interpretations and applications to statistical inference. *The American Statistician, DOI:10.1080/00031305.1993.10475930*, *47*(1), 33-38.

Everitt, B. (2005). *An R and S-plus companion to multivariate analysis*. Springer.

Ferrell, B. L. (1979). *Multivariate analysis*. Academic Press (Probability and Mathematical Statistics).

Gniazdowski, Z. (2013). Geometric interpretation of a correlation. *Zeszyty Naukowe Warszawskiej Wyzszej Szkoly Informatyki*(9, Rok 7), 27-35.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*(28 (3-4)), 321-377.

Izenman, A. J. (2008). *Modern multivariate statistical techniques: Regression, classification, and manifold learning*. Springer (Springer Texts in Statistics).

K. Liu, J. S., N. Gebraeel. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE T. Automation Science and Engineering*, *3*(10), 652-664.

Kolmogorov, A. (1936). Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari*(4), 83-91.

Ramasso, E., & Gouriveau, R. (2010). Prognostics in switching systems: Evidential Markovian classification of real-time neuro-fuzzy predictions. In *Ieee international conference on prognostics and system health management, macau, hong-kong, jan. 12-14.*

Randall, R. B. (2011). *Vibration-based condition monitoring - industrial, automotive and aerospace applications*. John Wiley and Sons, Ltd. Publications.

Rencher, A. C. (2002). *Methods of multivariate analysis*. Wiley Interscience (Wiley Series in Probability and Statistics).

Shewhart, W. A. (1931). *Economic control of quality of manufactured product.* David Van Nostrand.

Shewhart, W. A. (1986). *Statistical method from the viewpoint of quality control*. Dover Publications.

Timm, N. H. (2002). *Applied multivariate analysis*. Springer (Springer Texts in Statistics).

Tyurin, Y. N. (2009). Multivariate statistical analysis: A geometric perspective. *arXiv:0902.0408*.

Wickens, T. D. (Ed.). (1995). *The geometry of multivariate statistics*. Lawrence Erlbaum Associates, Incorporated.

W. K. Härdle, L. S. (2012). *Applied multivariate statistical analysis 3rd ed.* Springer.

An Aircraft Lifecycle Approach for the Cost-Benefit Analysis of Prognostics and Condition-based Maintenance based on Discrete-Event Simulation

Nico B. Hölzel[1], Thomas Schilling[2] and Volker Gollnick[3]

[1,2,3]*DLR - German Aerospace Center, Air Transportation Systems, Hamburg, 21079, Germany*

nico.hoelzel@dlr.de
thomas.schilling@dlr.de
volker.gollnick@dlr.de

ABSTRACT

The paper will provide a lifecycle cost-benefit analysis of the use of Prognostics and Health Management (PHM) systems and a conditioned-based maintenance (CBM) concept in future aircraft. The proposed methodology is based on a discrete-event simulation for aircraft operation and maintenance and uses an optimization algorithm for the planning and scheduling of CBM tasks. In the study, a 150-seat short-range aircraft equipped with PHM and subject to a CBM program will be analyzed. The PHM-aircraft will be compared with an Airbus A320-type of aircraft with maintenance expenditures equivalent to a conventional block check maintenance program. The analysis results will support the derivation of technical and economic requirements for prognostic systems and CBM planning concepts.

1. INTRODUCTION

Aircraft operators are under pressure to increase aircraft availability and operability in the future and continue to reduce the cost of aircraft operation. Reductions of maintenance downtimes and expenditures and the prevention of operational interruptions can help to achieve these objectives.

Technical and aircraft equipment was the most occurring direct delay category in 2006, with 10.2 % of total delays (Eurocontrol, 2007). When aiming for significantly higher reliabilities of future aircraft, it should be considered that 20 % to 50 % of all unscheduled removals are no-fault-founds (NFF) (Söderholm, 2007).

Prognostic concepts can positively influence the areas safety, maintainability, logistics, lifecycle costs, system design and analysis, and reliability of a product (Sun et al., 2010). There is a large potential for the reduction of overall life cycle costs of an aircraft by implementing comprehensive diagnostic and prognostic concepts (Roemer et al., 2001; Keller and Poblete, 2011; Scanff et al., 2007).

PHM may help to reduce operational interruptions due to unscheduled maintenance events, and maintenance downtimes due to (unnecessary) preventive maintenance. While significant advances in PHM systems are announced by industrial and academic research, several challenges have to be resolved for the onboard deployment of an aircraft-wide system (Sun et al., 2010). Besides the solving of technical issues one important prerequisite of an implementation is the provision of a reliable cost-benefit assessment of the onboard use of PHM. Such an analysis must be able to capture all relevant impacts of the technology on aircraft operation and maintenance over the aircraft lifecycle.

It has to be differentiated between general impacts, which can be also achieved through an installation of (retrofit) PHM systems in legacy aircraft, and wider impacts, which require extensive certification effort and/or the implementation of PHM during an early aircraft design stage.

In general, prognostic systems provide early detection of the precursor (and/or incipient) fault condition of a component and are capable to predict its remaining useful life (RUL) (Engel et al., 2000). In addition, the fault isolation and identification capabilities of PHM contribute to a reduction of no-fault-founds (NFFs) and support the trouble shooting process (Leao et al., 2007).

Further benefits require consideration of PHM in the certification phase or already in the aircraft design phase. Significant reductions in maintenance downtimes and costs can only be realized when a paradigm shift from periodic, preventive maintenance towards a predictive (i.e. condition-based) maintenance strategy takes place. The major

Nico Hölzel et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

expected benefits in this case are substitutions of preventive inspection tasks and reductions of waste of (component-) lives. This leads to reductions of overall maintenance cost and downtimes. These effects additionally influence spare parts pooling due to reduced spare parts demand and thereby allows a reduction in capital commitment (Hölzel et al., 2012).

Today´s maintenance programs are characterized by periodic, preventive and corrective tasks. While periodic tasks are foreseeable and easy to plan, time and effort for corrective work is more difficult to plan as they arise from the results of (preventive) inspections. With prognostics, many preventive inspections may become obsolete, while predictive tasks have to be planned and carried out with (potentially) short warning times. The increased planning complexity requires a different maintenance planning approach in order to achieve the aimed goals of a PHM and CBM implementation. Furthermore, CBM may lead to increasingly fluctuating demands for spare parts and new requirements to the maintenance supply chain.

The benefits, which can be realized in a specific application, depend on the current maintenance concept and the criticality of the monitored item in terms of safety and operational reliability of the aircraft. Therefore, a detailed modeling and analysis of all relevant factors and economic conditions is required.

2. GOAL OF STUDY

In general terms, this paper aims to facilitate informed decision making through the analysis and evaluation of PHM systems and CBM concepts in future aircraft. More specifically, it is the goal of this study to propose an appropriate method for analyzing the economic potentials of a PHM implementation in future aircraft in combination with a CBM planning concept. The applied methodology should be generic and feasible to analyze existing and future aircraft.

An approach is needed, that considers all phases in aircraft lifecycle and includes all relevant impacts of PHM systems and existing interdependencies with other elements of the air transportation system in a comprehensive way. In particular the selected approach has to consider the influence of a PHM use on aircraft operation. The use of a discounted cash-flow method is required to take into account the time value of money when assessing an aircraft over its entire lifecycle.

To consider uncertainties in component failure behavior, the methodology used in the study should be based on individual component failure probability functions. Performance levels (i.e. false alarm rates and missed failure rates) of PHM systems have to be included to account for imperfect sensors or prognostic algorithms. Previous analyses have shown that the prognostics performance level has a significant impact on the added value of a PHM system (Hölzel et al., 2012).

Furthermore, the selected approach should be able to model the operational and economic impacts of a CBM strategy. It should cover scheduled and unscheduled maintenance.

The approach is demonstrated in a case study to show the potential economic benefits of a PHM/CBM concept from an airline perspective.

3. METHODOLOGY

Economic assessments of PHM applications have been discussed by many authors (e.g. Banks et al., 2005; Feldman et al., 2009; Leao et al., 2007; Sandborn & Wilkinson, 2007; Scanff et al., 2007). Typical measures are lifecycle costs (LCC) or return-on-investment (ROI) estimates of the implementation costs and the potentials for cost avoidance (e.g. Banks et al., 2005). Leao et al. (2007) developed a cost-benefit analysis (CBA) methodology for PHM applied to (legacy) commercial aircraft. The method comprises a comprehensive set of equations for the quantification of benefits and costs, which are related to a PHM implementation. Their approach is capable to conduct assessments from an aircraft manufacturer's or operator's perspective, but it requires many inputs from technical analyses and PHM specialists. Sandborn and Wilkinson (2007) have proposed a lifecycle cost approach including a maintenance planning model based on discrete-event simulation. They consider various uncertainties with regard to PHM systems by using probability distributions as inputs for the model. The model provides a detailed picture of the usefulness of PHM on component or sub-system level, while it does not cover additional impacts and interactions on overall system (i.e. aircraft) level.

Both levels of analysis, component and overall system level, are needed, when a profound CBA of PHM with particular attention on the implementation of CBM should be provided. As outlined in section 2 the cost-benefit model must cover the relevant impacts of PHM on component or sub-system level and should consider the corresponding uncertainties. This component level must then be integrated on aircraft level, in order to simulate the effects of PHM and CBM in a realistic aircraft operation scenario.

The assessment approach presented in the paper is based on a discrete-event simulation of aircraft operation including a branch-and-bound algorithm for maintenance planning optimization. A lifecycle cost-benefit model evaluates the simulation results using a discounted cash-flow method. The presented simulation and assessment tool is modeled in MATLAB©. Aircraft type and operator specific XML-files are used to configure and control the lifecycle analyses.

3.1. Aircraft Lifecycle Approach

New technologies or concepts for the air transportation system need not only to lead to technological improvements, but also have to show economic advantages compared to the current system.

Direct operating cost (DOC) is an established metric to perform economic valuation of existing aircraft or future aircraft concepts. DOC formulae use global technical, operational, and economic parameters to come up with an average DOC value on a flight-cycle or flight-hour basis.

When assessing technologies and processes with impacts on the air transportation system level, all phases of the life cycle and interdependencies with other system elements have to be considered. New maintenance concepts influence maintenance cost and aircraft availability. To capture time and cost aspects, the lifecycle cost-benefit model AIRTOBS (Aircraft Technology and Operations Benchmark System) was developed.

initiated by the acquisition of an aircraft and ends with the decommissioning. The model includes aircraft specific parameters (e.g. acquisition cost, fuel consumption, seating capacity, crew size, and aircraft specific charges), operational aspects (e.g. route network, maintenance concepts and costs, and ticket prices), as well as global boundary conditions (e.g. fuel price trend, annual inflation rate). AIRTOBS focuses on the perspective of an aircraft operator and includes methods to account for costs and revenues.

An overview of AIRTOBS is shown in Figure 1. It consists of three main modules. The Flight Schedule Builder (FSB) generates a generic aircraft lifecycle flight schedule based on airline route data assuming full aircraft availability (i.e. no maintenance). Routes are considered based on the aircraft cycle time including flight time, taxi and runway operation times, and turnaround time.

This provisional flight schedule serves as the fundament for the Maintenance Schedule Builder (MSB). The MSB

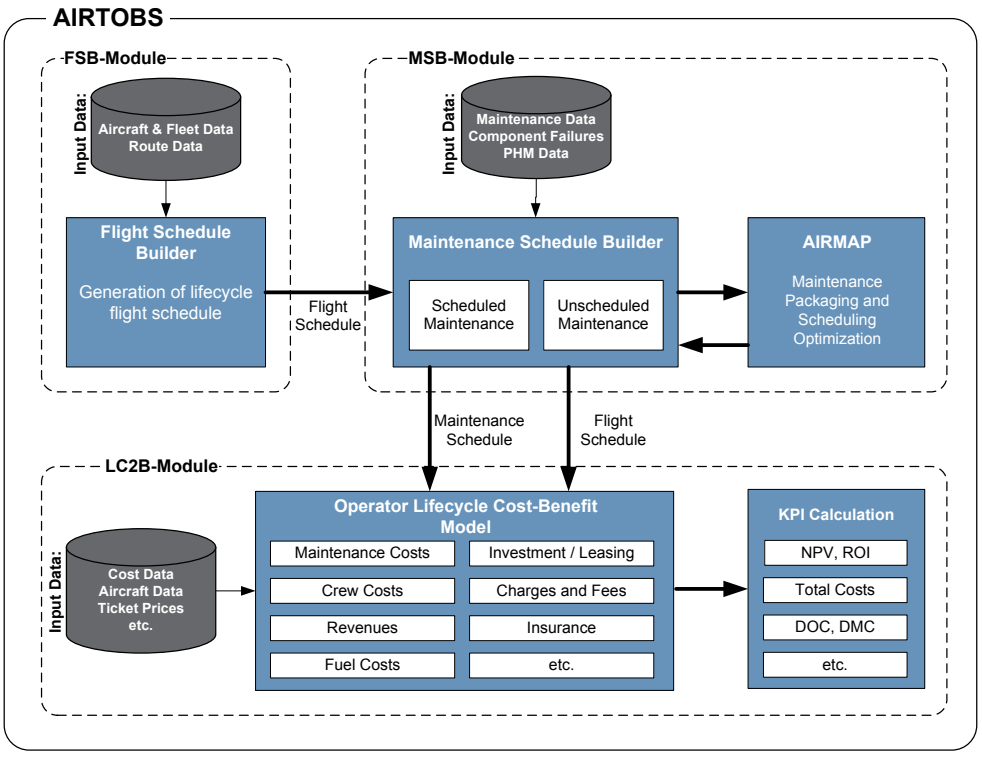


Figure 1. Lifecycle cost-benefit model.

The model is generic in nature and is feasible for economic assessments of various aircraft technologies and operation concepts from an operator's perspective. Apart from the assessment of prognostic concepts (Hölzel et al., 2012), studies on aircraft with natural laminar flow (Wicke et al., 2012) or intermediate stop operation concepts (Langhans et al., 2010) have been conducted.

It models all economic relevant parameters along the aircraft life cycle. The aircraft operational lifecycle is

executes a simulation run of the flight operation and maintenance events over the aircraft lifecycle. The MSB uses input data from maintenance databases for the modeling of scheduled and unscheduled maintenance events, including airframe, engine and component maintenance.

To analyze an application of PHM in combination with a CBM planning concept, a task-oriented maintenance modeling is used for the corresponding maintenance

activities. A maintenance packaging and scheduling optimization (AIRMAP) module (outlined in section 3.3) allocates maintenance tasks to maintenance events in a way that minimizes overall maintenance cost while ensuring that all scheduled flights can be carried out.

After the optimized maintenance schedule and the adjusted flight schedule are generated, the results are passed on to the Operator Lifecycle Cost-Benefit Model (LC2B), where costs and revenues are calculated. The actual time of occurrence of the cost and revenue elements is captured to account for the time value of money. All values are escalated over the aircraft lifecycle to account for inflation, before they can be summarized as net present value (NPV). It can be calculated as given in Eq. (1), where C_0 is the initial investment (i.e. aircraft price) and C_i is the cash-flow in the i-th year. The discount rate r represents the rate of return that could be achieved with equivalent investment alternatives in the capital market (Brealey, Myers, & Franklin, 2006). In business practice, a company or industry weighted average cost of capital (WACC) is often used as discount rate.

$$NPV = -C_0 + \sum_i \frac{C_i}{(1+r)^i} \quad (1)$$

The NPV is one among many other metrics that are calculated in AIRTOBS and can be used for the comparative evaluation of aircraft technologies and (operational) concepts.

3.2. Modeling of Maintenance Events and PHM Impacts

This section describes the modeling of maintenance events and the logic how the impacts of PHM on scheduled and unscheduled maintenance is implemented in the MSB module as depicted in Figure 1. The maintenance modeling is realized as discrete-event simulation based on the scheduled flights in aircraft lifecycle.

3.2.1. Scheduled Maintenance

Scheduled maintenance is considered depending on discrete, interval-based events. Intervals are specified by flight hours (FH), flight cycles (FC), and calendar time (years, months, days). Each event has a specific ground time, during which the flight schedule is adjusted while producing time discrete costs to the airline. To account for operating experience and maturity effects in maintenance, maturity curves are provided within the model. The maintenance schedule created by the MSB follows (by default) a traditional block check concept for line and base maintenance.

3.2.2. Unscheduled Maintenance

In order to model unscheduled maintenance, one must have knowledge of the failure behavior of the respective components or systems. This is achieved by using non-parametric failure distribution functions, which have been calculated on the basis of historic maintenance data. Particularly in order to attain feasible computing times in the following simulation process and to guarantee an appropriate size of the random sample, one distribution function was calculated for any component within ATA Chapters with identical first three digits (ATA 3D Chapter, i.e. subsystem level) (Hölzel et al., 2012).

Using the previously created lifetime flight schedule, unscheduled events are simulated based on component failure behavior, aircraft related mean times to repair (MTTR) and maintenance man-hours, e.g. downtime and man-hours needed for replacement of a component or LRU. In detail, the MSB module uses component lifetimes randomly drawn from previously described failure distribution functions. NFF events are modeled based on the NFF probabilities per FH that have been calculated from in-service data. The occurrence of an NFF event leads to an unscheduled removal of a component. PHM false alarm events are modeled in the same way as NFFs (Hölzel et al., 2012).

Component failures produce costs for labor and material. Furthermore they can result in flight delays or cancellations depending on the minimum equipment list (MEL), the MTTR, and the planned aircraft turnaround time. Delays are modeled as a reduction in aircraft availability and a cost element that covers passenger compensations and accommodation. Unscheduled failures not meeting the MEL-conditions can cause a flight cancellation when the remaining availability is not adequate to execute all planned flights of the respective day. In addition, a certain delay time threshold can be defined, which enforces a cancellation when a delay exceeds the threshold.

To consider the influences of maintenance strategies and component reliabilities on spare parts provisioning, related inventory costs are modeled. Overall LRU inventory costs are modeled based on estimated component quantities to meet a desired service level and the total carrying cost (capital and inventory cost). The estimated component quantities are calculated based on the aircraft utilization, quantities per aircraft, mean times between unscheduled removals (MTBURs), repair turnaround times and fleet size (Khan et al., 1999).

3.2.3. Impacts of PHM

An implementation of prognostics in aircraft systems can lead to a variety of operational and economic benefits. The main capability of PHM is the provision of advanced warnings of failures. The following benefits deriving from this capability are in focus of this study:

1. Reduction of unscheduled events due to failures (and NFFs) of items/components.

2. Enabling CBM: Transition from preventive to condition-based maintenance measures.

The underlying effects of PHM on aircraft maintenance are modeled in different ways.

The impact of PHM on unscheduled events is modeled in the unscheduled maintenance module as described in section 3.2.2. Impending failures or NFFs that are successfully detected by the prognostic system no longer result in unscheduled events. While NFF events are assumed to be completely avoided by PHM impending failures result in CBM tasks. Those CBM tasks are subject to the maintenance planning process described in the following section 3.3. Since no diagnostic or prognostic system will operate completely perfect, it is necessary to consider possible prognostic failures in the model. Two types of prognostic failures are taken into account:

1. False alarm: Prognostic system detects an impending failure, although no failure is impending, or system reports impending failure early.
2. Missed failure: Prognostic system does not detect an impending failure or detects it late.

Each failure of an item that is initially covered by PHM can evolve into a missed failure with a certain probability. A missed failure event has the same consequences as a failure not covered by PHM. The probabilities of false alarm and missed failure events depend on the performance level of the PHM system and are input values of the model.

The potential impact of PHM on preventive, scheduled maintenance tasks depends on its task-code. Scheduled maintenance tasks can be assigned to a variety of different task codes (Airbus, 2007) as listed in Table 1. While tasks with some task codes could become redundant if a PHM system is used, prognostics have no influence on other scheduled tasks listed in the scheduled maintenance program (MPD).

For the sake of simplification and generalization, the task codes are summarized to six task code groups (TCG) within the model as shown in Table 2. TCG 1 to 3 reflect tasks, which are potentially redundant, if a PHM system covers the contained tasks. The model assumes that the prognostic system is able to automatically carry out a certain fraction of the check- or inspection-tasks in a continuous or non-continuous manner. The fraction of tasks covered by a PHM system can be adjusted with the task redundancy parameter P_{TR}. It is obvious that this parameter is depending on the overall PHM coverage rate, but it is not necessarily identical. The parameter P_{TR} implies that it is possible to eliminate the corresponding scheduled maintenance task from the MPD under consideration of certification requirements.

Table 1. Maintenance task codes.

Task Code	Definition
BSI	Borescope inspection
CHK	Check for condition, leaks, circuit continuity, check fluid reserve on item, check tension and pointer, check fluid level, check detector, check charge pressure, leak check/test.
DI	Detailed inspection
DS	Discard
FC	Functional check/test
GVI	General visual inspection
LU	Lubrication
OP	Operational check/test
RS	Remove for restoration
SDI	Special detailed inspection
SV	Drain, servicing, replenishment (fluid change)
TPS	Temporary protection system
VC	Visual check

Table 2. Task code groups and potential PHM impact.

Task code Group (TCG)	Included task codes	Potential impact of PHM
TCG 1	CHK, OP, FC	Task elimination
TCG 2	GVI	Task elimination
TCG 3	DI, SDI	Task elimination
TCG 4	SV, DS, RS	Interval escalation
TCG 5	Non-routine	Interval escalation
TCG 0	Non-routine / other	No impact

If a significant fraction of scheduled tasks can be eliminated through a PHM implementation, this reduces the total workload and potentially also the aircraft downtime of a maintenance check. Without special consideration of the minimum duration of certain tasks ("shortest path"), the influence of PHM on aircraft downtimes can be estimated as shown in Eq. (2).

$$t_{DT,new} = t_{DT,0}\left(1 - P_{TR} \cdot r_{routine} \cdot r_{TR}\right) \quad (2)$$

$t_{DT,new}$ resulting maintenance downtime

$t_{DT,0}$ maintenance downtime without PHM impact (reference case)

P_{TR} task redundancy parameter

r_{TR} ratio of routine tasks potentially redundant in case of PHM use

$r_{routine}$ ratio of routine task man-hours to complete man-hours of check

It is assumed that preventive maintenance tasks related to TCG 4 have to be carried out less frequently when the corresponding items are monitored by PHM. This means, the former limited service life of the item is extended

through the use of PHM depending on the actual condition. Since no component degradation models are available for this study, the influence of PHM on service life is modeled with the interval escalation parameter P_{IE}, which is assumed as input value and can be varied in a parameter variation.

In addition to routine activities, scheduled checks also comprise large amounts of non-routine tasks. Detected findings result in non-routine activities (i.e. repairs or replacements of the respective items), when the degradation may reach a critical state prior to the next preventive inspection. It is assumed that a certain part of these non-routine tasks can be conducted at a later time, the respective items are subject to a CBM strategy (and monitored by PHM). These tasks are summarized in TCG 5. The last task code group (TCG 0) includes non-routine (e.g. findings that are critical for flight safety and thus have to be repaired immediately) and other tasks (e.g. cabin refurbishments and paintings) to which a PHM system has no influence.

3.3. Condition-based Maintenance Planning

The planning of aircraft maintenance is the allocation of maintenance tasks (i.e. objects) that must be carried out on specific aircraft to maintenance capacities (i.e. bins). Combinatorial problems of this character are of higher complexity and are very similar to the elementary bin-packing problem (Fukunaga et al., 2007; Bohlin, 2010). Since the aircraft maintenance planning, as discussed in this paper, considers more variables and constraints as the "simple" bin packing problem, it is very likely to be NP-hard [1]. Although the problem might not be solved in polynomial time, solutions can efficiently be verified, e.g. by using a branch-and-bound algorithm (Korte et al., 2006; Schröder, 2011).

In this study, each ground time of an aircraft (turnaround times and overnight stays) is regarded as a maintenance opportunity. It is the goal to minimize aircraft maintenance costs and to utilize existing maintenance opportunities efficiently while aircraft rotation planning and limited maintenance capacities are considered. This is achieved by appropriate grouping of maintenance tasks, while considering technical (maintenance intervals or RULs determined by a PHM system) and organizational restrictions. The process of grouping of tasks is referred to as maintenance task packaging in the following. The packaging of tasks allows an efficient use of maintenance opportunities but leads to waste of life when items are maintained earlier than required or tasks are performed before due date. The cost of wasted life is calculated as described in Eq. (3).

$$c_i^w = \frac{t_i^{life} - t_i^{RUL}}{t_i^{life}} \cdot c_i^{task} \tag{3}$$

c_i^w cost for wasted life of task i

c_i^{task} cost for performing task i (labor, material, logistics)

t_i^{life} complete life or interval of task i

t_i^{RUL} RUL or remaining time until due date of task i at time of task execution

The maintenance planning problem can be formulated with the objective function and the related constraints described in Eqs. (4) to (16).

$$\min \sum_{i \in I} \left(c_i^{task} + c_i^w\right) + \sum_{j \in J} c_j^{opp} + \sum_{k \in K} c_k^{fixed} \tag{4}$$

$$\sum_{i \in I} d_i x_i^k \leq m_k \qquad \forall k \in K \tag{5}$$

$$\sum_{i \in I} \sum_{p \in P} x_i^k y_i^p \leq s_k \qquad \forall k \in K \tag{6}$$

$$\sum_{i \in I} \sum_{j \in J} l_j^k x_i^k z_i^j = l_k \qquad \forall k \in K \tag{7}$$

$$\sum_{i \in I} x_i^k q_i = q_{s,k} \qquad \forall k \in K, \forall sk \in Q_k \tag{8}$$

$$\max\left(u \cdot t_i^{max}, t_i^{last} + u \cdot t_i^{max,int}\right) \leq t_i \leq \max\left(t_i^{max}, t_i^{last} + t_i^{max,int}\right) \qquad \forall i \in I \tag{9}$$

$$t_k \leq t_i \qquad \forall i \in I, \forall k \in K \tag{10}$$

$$t_j^{min} \leq t_i \leq t_j^{max} \qquad \forall i \in I, \forall j \in J \tag{11}$$

$$d_i^{MTTR} \leq t_j^{max} - t_j^{min} \qquad \forall i \in I, \forall j \in J \tag{12}$$

$$t_i^{last} + t_i^{max,int} - t_{end} \leq t_{tranfer} \wedge t_i^{max} - t_{end} \leq t_{transfer} \qquad \forall i \in I \tag{13}$$

$$x_i^k \in \{0,1\} \qquad \forall i \in I, \forall k \in K \tag{14}$$

$$y_i^P \in \{0,1\} \qquad \forall i \in I, \forall p \in P \tag{15}$$

$$z_i^J \in \{0,1\} \qquad \forall i \in I, \forall j \in J \tag{16}$$

Definition of symbols:

i index for maintenance task to be performed

I set of maintenance tasks to be performed

j index for maintenance opportunity

J set of maintenance opportunities

k index for maintenance location

K set of maintenance locations

[1] NP-hard describes a class of problems in computational complexity theory.

p	index for aircraft (tail-sign)
P	set of aircraft
Q_k	set of capabilities at maintenance location k
d_i	man-hours required for task i
d_i^{MTTR}	mean time to repair for task i
q_i	aircraft type of task i
t_i	actual starting time of execution of task i
t_i^{max}	RUL or remaining time until due date of task i
u	minimum usage factor for all t_i^{max} $(0 \leq u \leq 1)$
c_j^{opp}	fixed cost for usage of maintenance opportunity j
l_j	place of maintenance opportunity j
t_j^{min}	beginning of maintenance opportunity j (arrival of aircraft)
t_j^{max}	end of maintenance opportunity j (departure of aircraft)
$t_i^{max,int}$	maximum time between two events of task i
t_i^{last}	date of last allocation of task i
t_{end}	end of period
$t_{transfer}$	length of time from which a task is transferred to the next planning period
c_k^{fixed}	fixed cost for usage of maintenance location k
l_k	place of maintenance location k
m_k	available man-hours at maintenance location k
$q_{s,k}$	capability s at maintenance location k
s_k	available maintenance slots at maintenance location k
t_k	earliest availability of maintenance location k
x_i^k	1, if i is performed at k; 0, otherwise
y_i^p	1, if i belongs to p; 0, otherwise
z_i^j	1, if i is performed at j; 0, otherwise

The objective function of the maintenance planning problem is depicted in Eq. (4). The sum of all costs for the execution of maintenance tasks within the current planning period should be minimized. Equations (5) to (16) comprise the constraints, which are considered for this study. Equation (5) limits the total man-hours that can be allocated at a maintenance location. The slot restriction in Eq. (6) defines that the number of aircraft allocated to a maintenance location must not exceed its number of available maintenance slots. Equation (7) ensures that place of the maintenance opportunity l_j is identical with the maintenance location l_k. The maintenance location k has to be capable (i.e. has to be certified and must have the necessary equipment) to perform task i (Eq. (8)). Equation (9) describes that the time of execution t_i of task i must not be later than t_i^{max} and not before the minimum lifetime utilization ut_i^{max}. In the case of a multiple assignment of the same task within one period, the execution time of the task must refer to the respective task**.** The location availability constraint Eq. (10) describes that the time of availability t_k of location k must not be later than the time of execution t_i of task i. Equation (11) defines that the execution of task i must take place during a ground time of the aircraft. The ground time of the aircraft must be at least as long as the MTTR of the longest task to be allocated (Eq. (12)). The constraint Eq. (13) ensures that a task is allocated in the current period if its remaining time t_i^{max} exceeds the end of the period by no more than the buffer time $t_{transfer}$. Equations (14) to (16) are binary decision variables that allocate a task i to a location k, an opportunity j, and an aircraft p.

The CBM planning function used for this study is implemented in the AIRMAP model, which is a sub module of AIRTOBS (as shown in Figure 1). AIRMAP uses an optimization approach that can be characterized as depth-first-search branch-and-bound algorithm. The resulting task packaging and maintenance scheduling process is illustrated in Figure 2. The figure shows due dates (marked with an "X") for a number of tasks ("Task 1" to "Task n") in two random periods in aircraft life. For each planning period, the algorithm searches for a cost-minimal maintenance plan in an iterative process. The resulting maintenance events are marked with vertical dotted lines. The distances between the time of an event and the due dates of the allocated tasks represent the waste of life (expressed in FH). Due to the limitation of maintenance capacities and individual costs and man-hours of the tasks, it can be feasible to allocate a task to an event other than the nearest (e.g. allocation of second due date of "Task 5" to "Event 2" in Figure 2).

It is possible that the optimizer cannot allocate tasks, which are due shortly after the beginning of a new period because of a lack of maintenance opportunities. To avoid this, the user of the optimizer can define a buffer period that forces the algorithm to allocate the respective tasks in the preceding period (e.g. the third execution of "Task 1" is allocated to "Event 3" in Figure 2).

In this study, preventive scheduled and condition-based maintenance activities are subject to the previously described maintenance planning optimization. The maintenance optimization is designed as a dynamic planning approach that responds to varying maintenance needs and airline operation during aircraft lifecycle. This is achieved by splitting the operating lifecycle into shorter planning periods (e.g. four weeks) that are run through sequentially. This approach seems to be more realistic compared to a single optimization covering the complete lifecycle. In addition, this procedure leads to a significantly reduced computation time due to the reduction of the optimization

problem. In theory, longer planning periods would lead to better solutions from a lifecycle perspective.

The optimizer plans maintenance events for planning periods sequentially (beginning with aircraft entry into service). The algorithm takes into account only those tasks that are due in the current planning period. All other tasks are moved to the next planning period.

- a (lifecycle) flight schedule,
- economic boundary conditions like fuel price, ticket prices, labor cost, etc.

Based on the specified PHM system and a selected aircraft the component failure analysis is performed. This analysis results in unscheduled events and failures covered by PHM, which occur in the operating lifecycle. In parallel, the

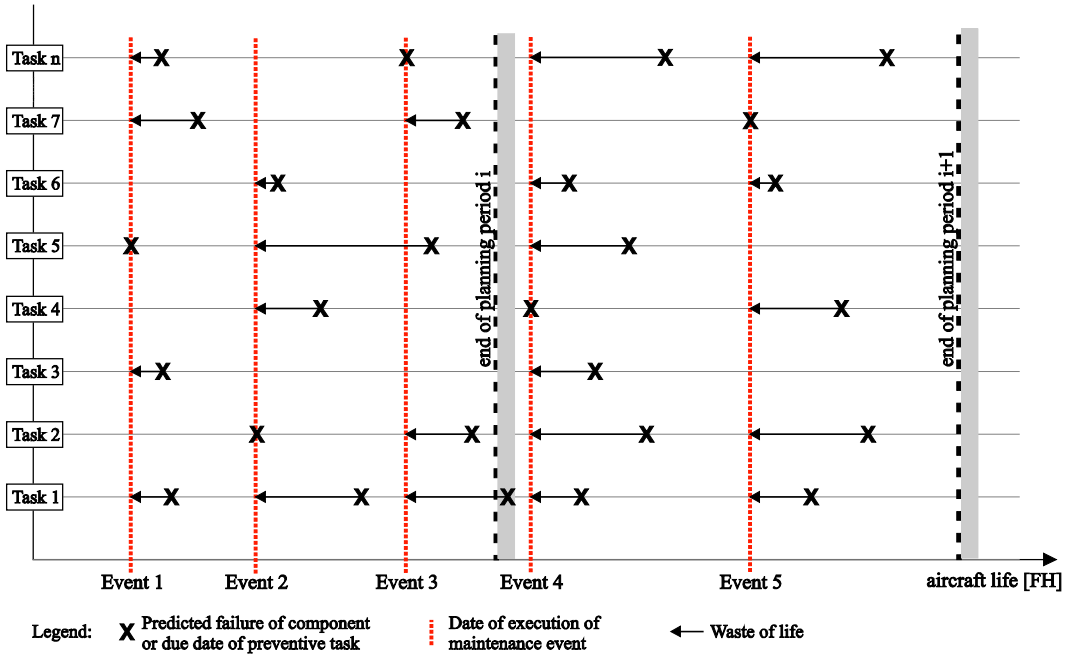


Figure 2. Maintenance scheduling and task packaging.

AIRMAP submits the best plan found to the Maintenance Schedule Builder (as depicted in Figure 1), which then generates the overall lifecycle maintenance and flight schedule as basis for the economic assessment in the LC2B module.

3.4. Assessment Approach

In the study, a 150-seat short-range aircraft equipped with PHM and subject to a CBM program will be analyzed and compared with the baseline. The baseline is formed by an Airbus A320-type of aircraft and a maintenance program equivalent to real world maintenance efforts in terms of man-hours (MH) and cost.

The economic analysis will follow the assessment approach as outlined in Figure 3. Required input data for the analysis are:

- the PHM concept to be analyzed, with specification of covered subsystems or components, corresponding prognostic performance levels and costs,
- a reference aircraft with its scheduled maintenance program, component failure behavior, DOC, etc.,

scheduled maintenance program is analyzed in terms of cost and man-hours efforts per task code. On this basis, a simplified maintenance program for the following analysis steps is modeled.

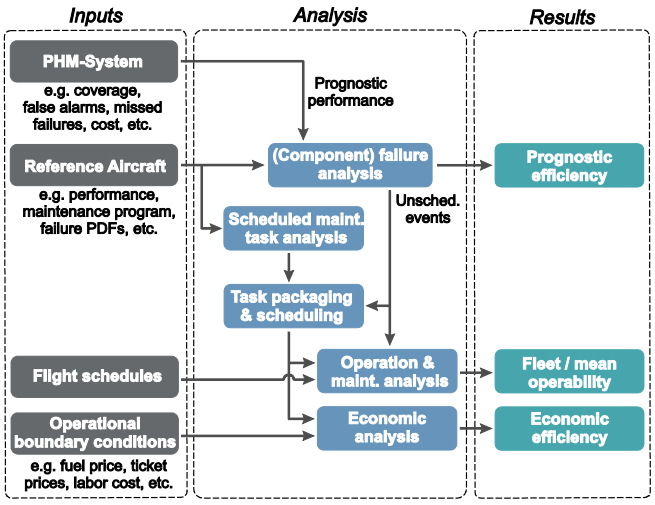


Figure 3. Assessment approach.

The maintenance scheduling and task packaging function then uses the results from both preceding steps und produces the optimized maintenance plan.

After that the analysis of aircraft operation and maintenance as well as the economic assessment are conducted using the AIRTOBS model.

Parametric studies will show the influences of prognostic performance levels, CBM implementation and maintenance planning constraints. From these studies, it is possible to derive essential requirements for prognostic systems and CBM concepts, e.g. minimum performance levels, maximal costs for acquisition and operation and minimum maintenance capacities, under given conditions.

4. ANALYSIS

The following analysis is intended to demonstrate that the proposed analysis approach is suitable to assess the overall benefits and costs of the use of PHM and CBM planning in aircraft lifecycle. While the results provide no answers regarding the suitability of specific PHM approaches or system architectures, they make it possible to derive technical and economic requirements for those in a subsequent step.

Studies following the proposed assessment approach require extensive data, which is usually – at least partially – considered confidential by airlines and maintenance, repair & overhaul (MRO) companies. For this reason, the authors have preferably used publicly available information only or have derived the required data under use of assumption from this information. The following section describes the essential data and the assumptions made for this study.

4.1. Data and Assumptions

An aircraft similar to an Airbus A320 will be used as a reference in this study. This applies to the typical aircraft operation, the maintenance program and all recurring and non-recurring costs as well as expected revenues in the operational lifecycle of this type of aircraft.

It is assumed that aircraft configurations used in this study have the same technology level as today's A320 aircraft, but with PHM installed.

The following sections describe the data and assumptions made for the aircraft operation, scheduled and unscheduled maintenance, and relevant operational boundary conditions.

4.1.1. Aircraft Lifecycle and Operations

An operating lifecycle of 25 years is assumed in this study. The aircraft is operated by a full-service network carrier on a short-range rotation with a daily utilization of 7.5 FH. Table 3 shows details of an assumed aircraft operation.

Table 3. Aircraft operational data.

Parameter	Unit	Value
Operating days/week	[d]	7
Night curfew	[h]	7
Flights per day	[FC]	6
FH/FC	-	1.25
Taxi time per FC	[h]	0.3
Turn-around time	[h]	0.75
Block fuel	[kg]	4,000

4.1.2. Scheduled Maintenance

The major part of the scheduled maintenance requirements for an aircraft is defined in the MPD. This manufacturer documentation contains maintenance tasks with specification of intervals and required man-hours that are to be carried out during service life. Maintenance cost data and more realistic estimates of the related man-hours are for example published by Aircraft Commerce (2006). These data describe traditional block check concepts as still followed by many aircraft operators today.

The intended transition from preventive to condition-based tasks in this study, however, requires an equalized or task-based approach. To enable a convincing CBA of PHM and CBM, it must not be mixed with a comparison between block check and equalized or task-based maintenance concepts.

This leads to the necessity that also the reference maintenance program needs to follow a task-based approach.

Table 4. Scheduled maintenance program A320 (derived from Aircraft Commerce, 2006).

Check	Down-time [h]	Interval	MH [h]	Material cost [US$]
Transit & Pre-flight	0	1 FC	2.6	7
Ramp Check	0	2 d	4	500
Service Check	0	7 d	10	700
A-Check	24	600 FH	80	5.5 k
C-Check	138	18 mo.	2,000	38 k
IL-Check	336	72 mo.	14,300	380 k
D-Check	672	144 mo.	20,000	1.5 M

Following this approach, a simplified task-based maintenance program has been modeled, which is equivalent to the real A320 maintenance program in terms of man-hours and cost as described in Table 4. The maintenance events outlined in Table 4 cover routine and non-routine tasks as well as cabin refurbishments and

typical volume of work resulting from Airworthiness Directives (AD) and Service Bulletins (SB).

The modeled reference maintenance program, referred to as equivalence maintenance program in the following, consists of two parts:

1. Task-based concept for short and medium interval tasks (former Service Check, A-Check, and C-Check),
2. Block checks for long interval tasks (former IL- and D-Check).

Transit & Pre-flight Checks can be performed at any airport and do not require an additional maintenance downtime. That is why these checks are not considered for the composition of an equivalence maintenance program and in the following maintenance planning and optimization process.

Analyses of the scheduled maintenance tasks contained in the A320 MPD result in the shares of the different task codes (as previously described in Table 1) shown in Figure 4. The derived man-hours shares have been clustered according to their interval lengths. For this purpose, a pragmatic division into short, medium and long intervals has been made. While the short and medium intervals correspond to the intervals of the former Service, A-, and C-Checks, the long intervals comply with the IL- and D-Check intervals. These values form the basis for the modeled routine tasks of the equivalence maintenance program.

While the MPD only consists of routine maintenance tasks, non-routine tasks account for a large part of overall maintenance expenditures. It is assumed for this study that there are non-routine tasks that could be performed at a later time, if a PHM (or structural health monitoring) system monitors the health state of the respective item (e.g. cracks in a structural component, which are not critical at the time of discovery).

However, there are non-routine and other maintenance tasks, which are not influenced by PHM at all (e.g. repairs or removals of faulty items, cabin overhauls, painting, or tasks resulting from ADs or SBs). Since no detailed breakdown of non-routine workload could be determined, the ratio of TCG-5 to TCG-0 is assumed as 50:50 in the following.

The allocation of short and medium interval man-hours to their respective TCGs results in the first part of the equivalence maintenance program shown in Table 5.

The modeled equivalence maintenance program consists of 12 short interval and 71 medium interval tasks, which represent the maintenance man-hours and task code groups shown in Table 5 over the lifecycle of 25 years. The short interval tasks are characterized by intervals between 80 and 1000 FH. The intervals of the medium interval tasks range from 4,500 to 13,500 FH.

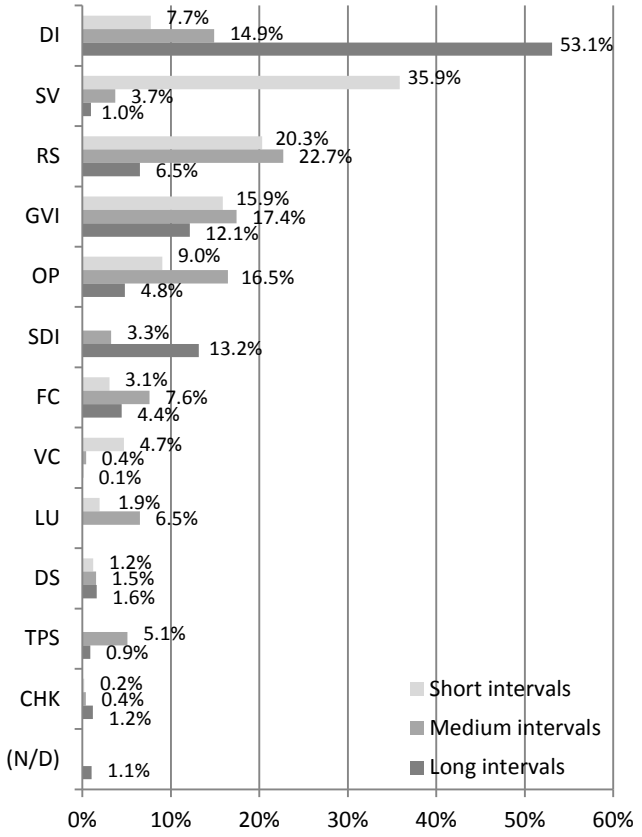


Figure 4. Distribution of man-hours over task codes in 12-year-period.

It is assumed that the 6- and 12-year heavy maintenance checks (former IL-/D-check) will persist as block check events. As a consequence, an interval extension of one task of a heavy maintenance check does not lead to an interval escalation of the total check, unless the intervals for all tasks of the checks are being extended accordingly.

Table 5. Equivalence maintenance program – Part 1 (equalized check events).

	TCG	Short interval		Medium interval	
		MH	Ratio	MH	Ratio
Routine	1	1,902	8.4 %	3,298	11.0 %
	2	2,454	10.8 %	2,355	7.9 %
	3	1,193	5.3 %	2,453	8.2 %
	4	8,881	39.2 %	3,773	12.6 %
Non-routine	5	3,588	15.9 %	8,250	27.5 %
	0	4,612	20.4 %	9,871	32.9 %
	Sum	**22,630**	**100 %**	**30,000**	**100 %**

Analysis of long interval tasks (6-/12-year check tasks and other tasks with intervals longer than generic C-check interval) show that about 89 % account for TCG 1 to 3,

which could be subject to task elimination. Only 9 % of the tasks account for TCG 4, which could be subject to interval escalation. The following analysis considers in connection with the block check events only the potential PHM impact of task redundancy, which accounts for almost 90 % of the routine work. The part 2 of the modeled equivalence maintenance program is summarized in Table 6.

Table 6. Equivalence maintenance program – Part 2 (remaining block check events).

	TCG	IL-Check		D-Check	
		MH	Ratio	MH	Ratio
Routine	1	941	89 %	1,568	89 %
	2	1,092		1,820	
	3	5,963		9,938	
	4	821	9 %	1,368	9 %
	other	183	2 %	305	2 %
	Sum	**9,000**	**100 %**	**15,000**	**100 %**
Non-routine	5	2,500	50 %	4,250	50 %
	0	2,500	50 %	4,250	50 %
	Sum	**5,000**	**100 %**	**8,500**	**100 %**

The applied generic modeling approach allows the comparison of a current maintenance program with any potential or future maintenance program without having described all maintenance tasks precisely. Particularly in early design stages of new aircraft, the proposed methodology could be beneficial in order to estimate the impact of alternative maintenance concept early on.

4.1.3. Unscheduled Maintenance

The modeling of unscheduled maintenance events in this study follows the approach as described in section 3.2.2. A total of 25 aircraft subsystems are considered in the study. The failure behavior of each subsystem is described by an individual non-parametric failure distribution function. It is assumed, that 12 of the 25 subsystems are potential candidates for a PHM implementation with a PHM coverage ranging from 0 to 100 percent. This means for the following analysis: A theoretical PHM-coverage of 100 % corresponds to a detection and prediction of all impending failures of the 12 selected subsystems. To limit the computing times, the PHM coverage rates for each of the 12 subsystems are assumed to be identical in all analyses.

4.1.4. Operational Boundary Conditions

In order to be able to evaluate the monetary results, a summary of the relevant economic data used in the analysis is given in Table 7. Assumed ticket prices for economy (EC) and business class (BC) influence airline revenues in the lifecycle CBA. The initial investment cost C_0 is assumed as 50 Mio. US$ (aircraft list price in 2008 less an assumed price discount of 35 %). This study should not provide cost estimates for the development and implementation of PHM systems. Rather, the goal is to derive maximum acceptable investment costs for PHM systems from the analysis results. Therefore, no additional fix costs for an airplane equipped with PHM are considered.

The delay costs of 0.63 US$ per passenger per minute include costs of passenger compensation and rebooking for missed connections, but also considers the costs of potential loss of revenue due to future loss of market share as a result of lack of punctuality (Eurocontrol, 2007). The internal rate of return r, which is used for the discounted cash-flow calculation, is assumed at 7 %.

Table 7. Summary of economic and operational data.

Parameter	Unit	Fiscal year	Value
Ticket price - EC	[US$]	2008	111
Ticket price - BC	[US$]	2008	334
Aircraft price C_0 (incl. 35% discount)	[Mio. US$]	2008	50
Labor rate (maintenance)	[US$/MH]	2009	70
Fuel price (fuel price scenario)	[US$/gal]	2013	2.49
Delay cost	[US$/min/pax]	2009	0.63
Average inflation	[1/year]		0.02
Discount rate r	[-]		0.07

4.2. Parameter Variation

Since the PHM and CBM concepts to be evaluated in this study are not implemented in commercial aircraft yet, actual performance characteristics of such concepts on aircraft level can hardly been estimated today. In addition, as mentioned previously, the proposed assessment methodology should provide assistance in the early design stage of future PHM and CBM concepts. For these reasons, it seems to be necessary to conduct a variation of parameters that characterize the performance of such concepts.

To limit the number of analyses and resulting calculation times in this study, three parameters are selected for the variation. These are “PHM coverage”, “task redundancy” and “interval escalation”. The parameters and their values are depicted in Table 8. The PHM coverage rate describes the portion of failures for which a specific prognostic system can report imminent failures, without consideration of false alarms and missed failures (see also section 4.1.3). The task redundancy rate is the percentage of preventive maintenance tasks that can potentially be eliminated if a PHM system is used to monitor the respective item (see also section 3.2.3). The interval escalation rate describes the factor by which preventive maintenance intervals may be extended if the corresponding item is monitored by a PHM system.

Table 8. Parameter space for analysis.

Parameter		Values
P_{Cov}	PHM coverage	0 0.25 0.5 0.75 1
P_{TR}	task redundancy	0 0.1 0.2 0.3..... 1
P_{IE}	interval escalation	0 0.25 0.5 0.75 1

The parameter space as defined in Table 8 results in 275 separate analyses, which have been conducted. In this study, each analysis consists of 100 simulation runs (Monte Carlo simulations) to account for the probabilistic behavior of the unscheduled maintenance module (due to the probabilistic modeling of the component failure behavior and the impact of PHM). Although a larger number of simulations might be desirable, the number had to be limited here to provide acceptable computing times.

4.3. Analysis Results

The performed analysis provides technical-operational and economic results. All results describe values for the operative lifecycle on a single aircraft. Since the study comprises 275 separate lifecycle analyses, only a limited selection of results can be presented in this paper.

Figure 5 and Figure 6 show the impacts of a variation of the parameters P_{TR} and P_{IE} on man-hours for maintenance tasks planned in AIRMAP. The absolute level of man-hours at $P_{Cov} = 1$ (Figure 6) is about 8,000 hours higher (over the lifecycle) than at $P_{Cov} = 0$ (Figure 5). The component maintenance events covered by PHM are responsible for this different level of man-hours. The shape of the curves is very similar in both cases.

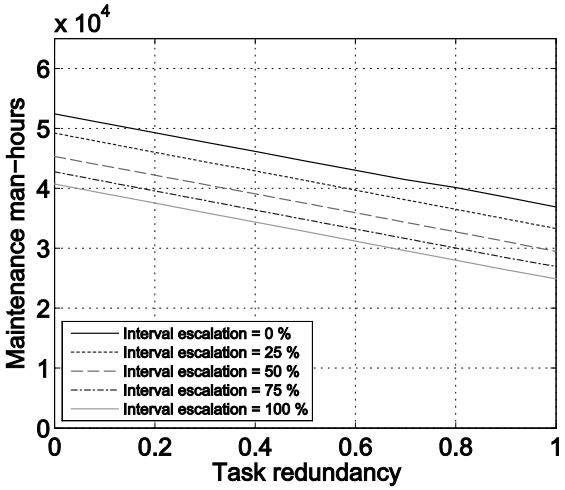


Figure 5. Man-hours for AIRMAP-tasks ($P_{Cov} = 0$).

As discussed in the beginning, a central goal of a PHM and CBM implementation is to improve the aircraft availability in order to increase the utilization. Both effects, the reduction of unscheduled events and the elimination of tasks, can contribute to higher aircraft utilization. Figure 7 shows that – even without a change in the aircraft operation concept – up to 420 additional flight cycles could be realized in aircraft lifecycle.

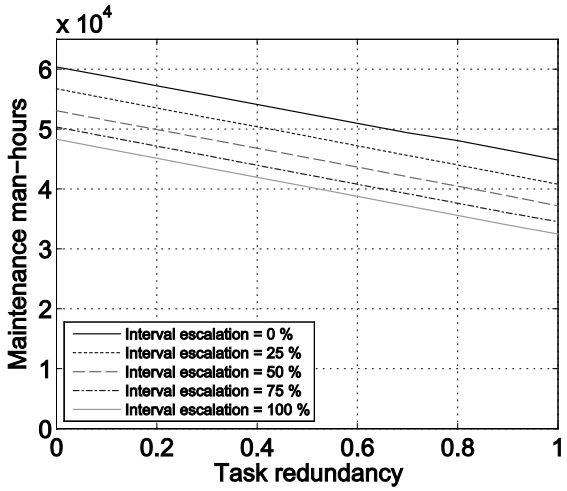


Figure 6. Man-hours for AIRMAP-tasks ($P_{Cov} = 1$).

Under the assumptions of this study, the avoidance of unscheduled events enables up to 260 additional flight cycles. Another 160 flights can be realized by shortening the maintenance downtimes for IL- and D-Checks in case of $P_{TR} = 1$.

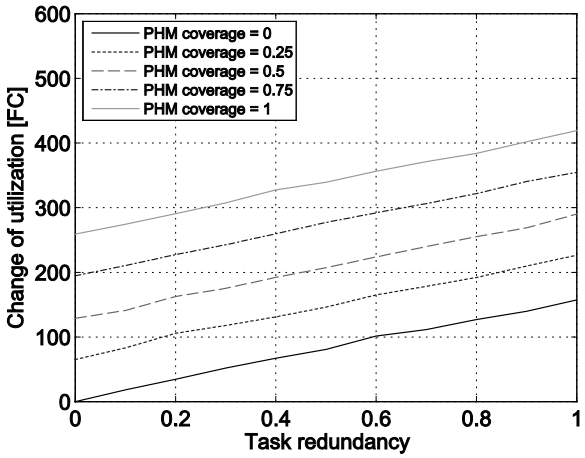


Figure 7. Aircraft utilization.

Figure 8 shows the impact of PHM coverage on the different categories of maintenance cost with the resulting changes of airline revenues and NPV. Since P_{TR} and P_{IE} are zero the figure shows the isolated benefit of the reduction of unscheduled events.

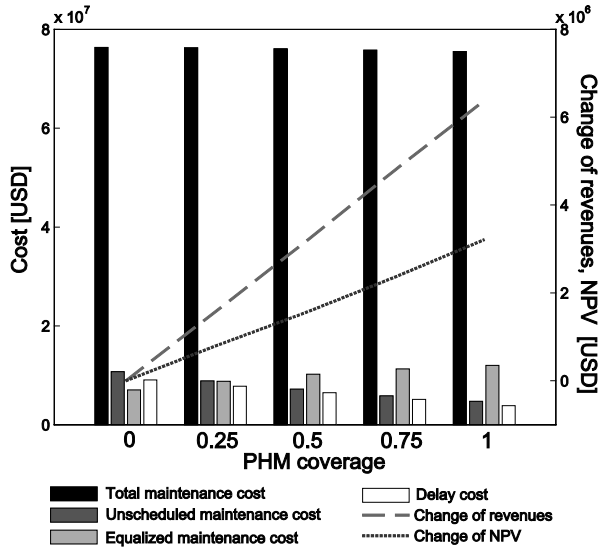


Figure 8. Impact of PHM on cost, revenues, and NPV (with $P_{TR} = 0$, $P_{IE} = 0$).

While total maintenance cost remains almost constant, a transition of unscheduled maintenance to dynamically planned, equalized maintenance (i.e. maintenance tasks planned in AIRMAP) can be observed for increases of PHM coverage. Moreover, the delay cost (which are not included in total maintenance cost) decreases significantly by almost 60 %. The reductions of unscheduled events lead to maximum increase of revenues of 6.3 million USD, which results in a higher NPV of 3.2 million USD (for $P_{Cov} = 0$).

The isolated influence of a variation of P_{TR} and P_{IE} on total maintenance cost is shown in Figure 9, when $P_{Cov} = 0$. The benefit of an escalation of task intervals can account for a cost reduction of 1.3 million USD ($P_{IE} = 1$).

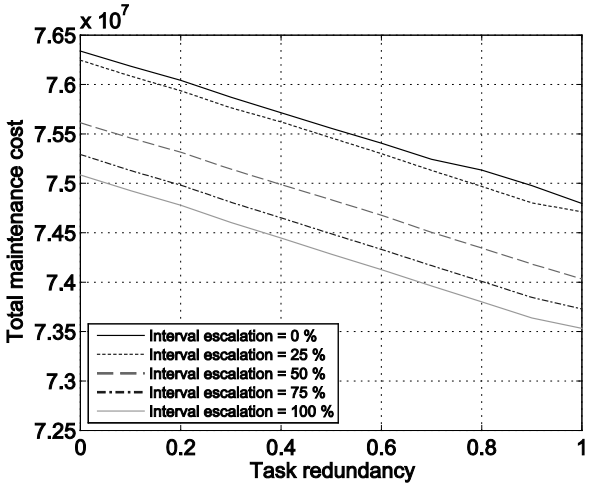


Figure 9. Total maintenance cost ($P_{Cov} = 0$).

Figure 10 shows the respective effect on total maintenance cost, when $P_{Cov} = 1$. It can be seen that the curves are principally shifted vertically to lower maintenance cost compared to Figure 9.

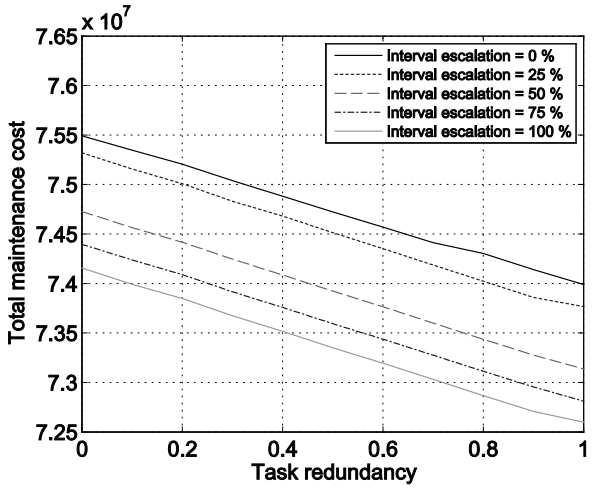


Figure 10. Total maintenance cost ($P_{Cov} = 1$).

Figure 11 describes the highest aggregated economic results of the presented study. The monetary benefit of an aircraft operator, expressed as NPV, is shown for all variations of P_{Cov}, P_{TR}, and P_{IE}. Each of the five parts of Figure 11 shows the impacts of the task redundancy rate and the interval escalation factor on airline NPV with the respective PHM coverage rate. It can be seen that the maximum benefit of an interval escalation (i.e. the difference of NPV for P_{IE} =0 % and P_{IE} =100 % in each subfigure) accounts for around 0.5 million USD. The maximum overall increase of NPV that could be realized under given assumptions is 4.75 million USD (as depicted in Figure 11 e). Although it is unlikely that a PHM-coverage of 100 % for the selected systems could be achieved at an acceptable price, the results show the range of potential benefits. The increase in NPV by a certain PHM/CBM configuration is at the same time the upper limit of the acquisition cost of such a system, which could be accepted.

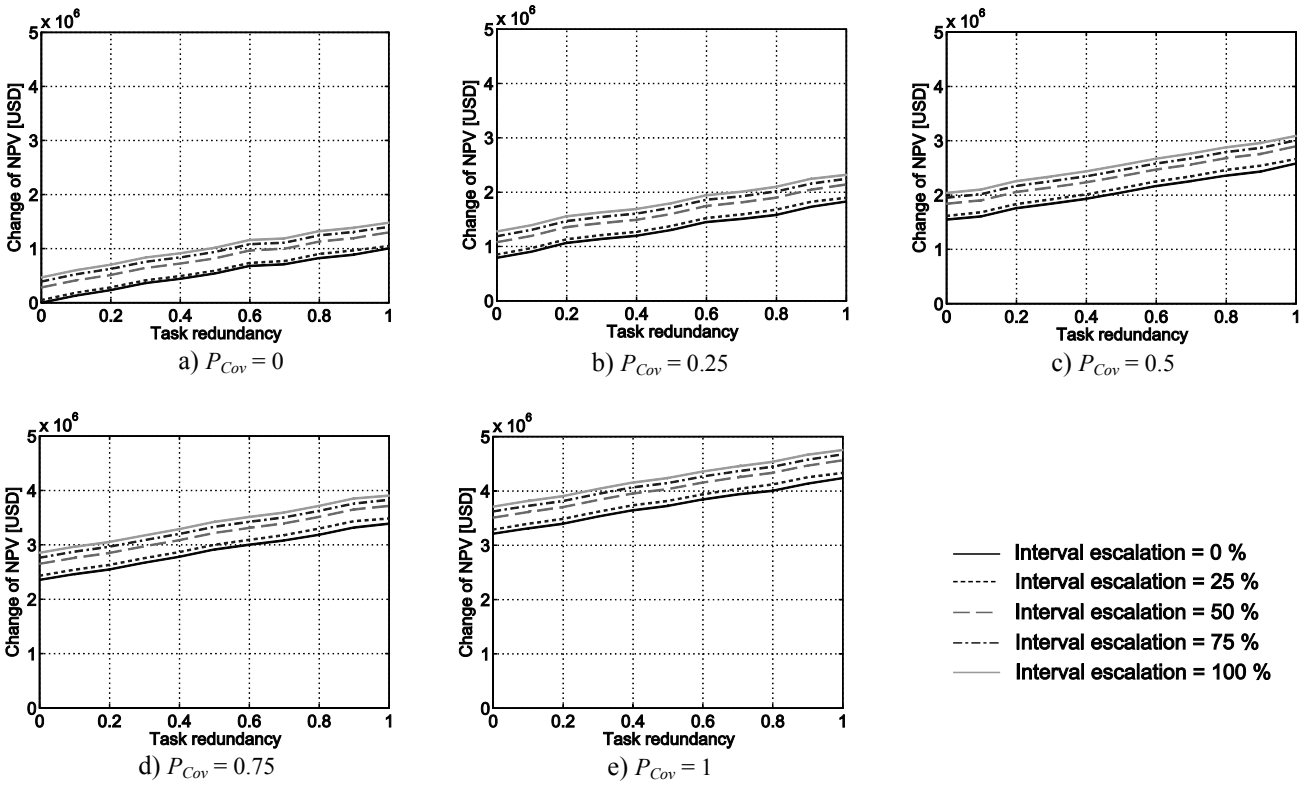


Figure 11. Impact of PHM coverage, task redundancy, and interval escalation rates on NPV.

The results presented in this section are on single aircraft level. The analysis does not consider interdependencies between different aircraft in a fleet. While AIRMAP is able to conduct the maintenance planning optimization for a fleet of aircraft, the other modules of AIRTOBS can only handle single aircraft at present.

5. CONCLUSION AND OUTLOOK

In this paper we have presented an integrated approach to model the impacts of PHM and CBM planning from an aircraft lifecycle perspective. The integration of the CBM planning approach in a lifecycle cost-benefit model allows the economic assessment of a PHM and CBM implementation in future aircraft. The application of the assessment approach can deliver valuable requirements for the future development of PHM and CBM concepts and demonstrate its consequences for operators and MROs.

At present, the assessment approach is limited to a single aircraft analysis. An extension of AIRTOBS on a fleet-level basis would allow using the complete functional range of AIRMAP, i.e. scheduling maintenance tasks and planning capacities for a fleet of different aircraft types on an airline's network. It is expected that an analysis on a fleet-level will result into a lower economic benefit per aircraft. This is because several aircraft compete for limited maintenance resources, leading to less efficient solutions of the CBM planning process.

In further studies we intend to analyze the effects of varying daily aircraft utilizations in order to investigate the applicability and benefits of the approach for different airline business models (e.g. network or low-cost carrier). Low-cost carriers usually have significantly higher aircraft utilizations and therefore shorter and less maintenance opportunities compared to a network carrier operating a similar route network. This fact may imply a higher sensitivity to flight schedule disturbances and consequently also a greater benefit from the reduction of unscheduled events due to the use of PHM. In contrast, decreasing aircraft ground times make it more difficult to solve the CBM planning problem and potentially reduce the efficiency of the maintenance plan.

Further improvements of the optimization algorithm included in AIRMAP in terms of computation times would allow analyzing significantly larger parameter spaces and a higher number of Monte Carlo simulations in the future.

ACKNOWLEDGEMENT

This study was carried out as part of the work in OMAHA, a research project supported by the Federal Ministry of Economic Affairs and Energy in the national LuFo V program. Any opinions, findings and conclusions expressed

in this document are those of the authors and do not necessarily reflect the views of the other project partners.

Supported by:

on the basis of a decision
by the German Bundestag

NOMENCLATURE

AD	Airworthiness Directive
AIRTOBS	Aircraft Technology and Operations Benchmark System
ATA	Air Transport Association
BC	business class
CBA	cost-benefit analysis
CBM	condition-based maintenance
DOC	direct operating cost
EC	economy class
FC	flight cycle
FH	flight hour
FSB	Flight Schedule Builder
LC2B	Life Cycle Cost-benefit Model
LCC	life cycle cost
LRU	line replaceable unit
MEL	minimum equipment list
MH	man-hours
MRO	maintenance, repair, and overhaul
MSB	Maintenance Schedule Builder
MTTR	mean time to repair
MTBUR	mean time between unscheduled removals
NFF	no fault found
NP	non-deterministic polynomial-time
NPV	net present value
PHM	Prognostics and Health Management
ROI	return on investment
RUL	remaining useful life
TCG	task code group
SB	Service Bulletin
XML	Extensible Markup Language

REFERENCES

Aircraft Commerce (2006). A320 family maintenance analysis & budget. *Aircraft Commerce*, issue 44, February/March 2006, pp. 18-31.

Airbus (2007). *A318/A319/A320/A321 Maintenance Planning Document*, rev. 30, May 01/07.

Banks, J., Reichard, K., Crow, E., & Nickell, K. (2005). How engineers can conduct cost-benefit analysis for PHM systems. *Proceedings of IEEE Aerospace Conference*, March 5-12, Big Sky, MT. doi: 10.1109/AERO.2005.1559701

Bohlin, M. (2010). *A Study of Combinatorial Optimization Problems in Industrial Computer Systems*, Doctoral dissertation, Mälardalen University, Sweden. www.es.mdh.se/pdf_publications/1622.pdf

Brealey, R., Myers, S., & Franklin, A. (2006). *Corporate Finance*, 8th edition, New York: McGraw-Hill Irwin.

Engel, S. J., Gilmartin, B. J., Bongort, K., & Hess, A. (2000). Prognostics, the real issues involved with predicting life remaining. *Proceedings of IEEE Aerospace Conference*, March 18-25, Big Sky, MT. doi: 10.1109/AERO.2000.877920

Eurocontrol (2007). A Matter of Time: Air Traffic Delay in Europe. *Eurocontrol Trends in Air Traffic*, vol. 2.

Feldman, K., Jazouli, T., & Sandborn, P. A. (2009). A Methodology for Determining the Return on Investment Associated With Prognostics and Health Management. *IEEE Transactions on Reliability*, vol. 58, no. 2, pp. 305-316.

Fukunaga, A. & Korf, R. (2007). Bin Completion Algorithms for Multicontainer Packing, Knapsack, and Covering Problems. *Journal of Artificial Intelligence Research*, vol. 28, pp. 393-429.

Hölzel, N. B., Schilling, T., Neuheuser, T., & Gollnick, V. (2012). System Analysis of Prognostics and Health Management Systems for Future Transport Aircraft. *Proceedings of the 28th Congress of the International Council of the Aeronautical Sciences (ICAS)*, September 23-28, Brisbane, Australia.

Keller, K., & Poblete, J. (2011). The Business Case for SHM, *System Health Management: with aerospace applications*, edited by Johnson, S. B. et al., Wiley, Chichester, United Kingdom, pp. 77-91.

Khan, K. A., & Houston, G. D. (1999). Design Optimization using Life Cycle Cost Analysis for Low Operating Costs. *RTO AVT Specialists' Meeting on Design for Low Cost Operation and Support*, held in Ottawa, Canada.

Korte, B., & Vygen, J. (2006). *Combinatorial Optimization*, 3rd edition, Berlin/Heidelberg, Germany.

Langhans, S., Linke, F., Nolte, P., & Schnieder, H. (2010). System analysis for future long-range operation concepts. *27th Congress of the International Council of the Aeronautical Sciences (ICAS)*, September 19-24, Nice, France.

Leao, B. P., Fitzgibbon, K. T., Puttini, L. C., & Melo, G. P. B. de (2008). Cost-benefit analysis methodology for PHM applied to legacy commercial aircraft. *Proceedings of IEEE Aerospace Conference*, March 1-8, Big Sky, MT. doi: 10.1109/AERO.2008.4526599

Roemer, M., Byington, C., Kacprzynski, G., Vachtsevanos, G., & Goebel, K. (2011). Prognostics. *System Health Management: with aerospace applications*, edited by Johnson, S. B. et al., Wiley, Chichester, United Kingdom, pp. 281-295.

Sandborn, P. A., & Wilkinson, C. (2007). A maintenance planning and business case development model for the application of prognostics and health management (PHM) to electronic systems. *Microelectronics Reliability*, vol. 47, no. 12, pp. 1889-1901.

Scanff, E., Feldman, K. L., Ghelam, S., Sandborn, P., Glade, M. & Foucher, B. (2007). Life cycle cost impact using prognostic health management (PHM) for helicopter avionics, *Microelectronics Reliability*, vol. 47, pp. 1857-1864.

Söderholm, P. (2007). A system view of the No Fault Found (NFF) phenomenon. *Reliability Engineering and System Safety*, vol. 92, pp. 1-14.

Schröder, C. (2011). *Erstellung eines Algorithmus für die Optimierung der Flugzeuginstandhaltungsplanung*. Diploma thesis, Institute of Air Transportation Systems, Hamburg University of Technology, Hamburg, Germany.

Sun, B., Zeng, S., Kang, R. & Pecht, M. (2010). Benefits analysis of prognostics in systems, *2010 Prognostics & System Health Management Conference*, January 12-14, Macao.

Wicke, K., Kruse, M. & Linke, F. (2012). Mission and economic analysis of aircraft with natural laminar flow, *28th Congress of the International Council of the Aeronautical Sciences (ICAS)*, September 23-28, Brisbane, Australia.

BIOGRAPHIES

Nico B. Hölzel is a research engineer at the German Aerospace Center (DLR) within the Institute of Air Transportation Systems located at the Hamburg University of Technology, Germany. Since February 2009, he is a member of the system analysis group and works on the operational and economic assessment of PHM technologies and aircraft maintenance concepts. From 2006 to 2009, he was a Senior Consultant in the automobile industry in Germany. He received his Diploma degree in Industrial Engineering and Management from Hamburg University of Technology in 2006.

Thomas Schilling is a research engineer at the German Aerospace Center (DLR) within the Institute of Air Transportation Systems located at the Hamburg University of Technology, Germany. As a member of the system analysis group he focuses on the assessment of aircraft maintenance and operational aspects within the air transportation system. He worked for two years for Airbus, Cabin & Cargo Customization. He holds a BSc in General Engineering Science, Aircraft Systems Engineering and a diploma in Mechanical Engineering, Theoretical Mechanical Engineering from Hamburg University of Technology.

Volker Gollnick holds the chair and directorate of the Institute for Air Transportation Systems of the German Aerospace Center (DLR) at the Hamburg University of Technology (TUHH). He holds a Diploma degree in Mechanical and Aerospace Engineering from the Technical University of Braunschweig and a Ph.D. degree from the Technical University of Munich. In his research he is in particular interested in operational improvements and changes of sustainable but efficient air transportation systems. Integration and collaboration of the various stakeholders is his special focus, when he is researching on new aircraft, airports and aircraft operations. He represents more than 20 years experience in aerospace industry and government. He held responsible roles in particular in flight testing, overall aircraft design, cockpit and avionics systems at EADS and its subsidiaries and the German Forces Flight Test Center.

A Reference Stack for PHM Architectures

Charles Crabb

LogTech, LLC, Wall, NJ, 08736, USA

ccrabb@logtech.com

ABSTRACT

This paper suggests a reference model for PHM processes that aids the customer of PHM in developing a business case for adopting PHM in his or her supply chain. Various PHM systems have been envisioned and developed in order to produce a prognosis of system or component behavior by collecting physical data from some section of a system, analyzing it and reporting the results to the entity that benefits from it, notably the supply chain that manages the components and receives the resulting cost benefit from PHM. All these systems have varying configurations that involve the collection of different types of data in different ways, the analysis of varying types of physical behavior and have different types of customers (different supply chain configurations). The customer needs to include the cost and complexity of the PHM system in his or her business model but has no formal standard to determine bounds on the complexity of the PHM system. Just as there are reference stacks for service-oriented architectures, this paper proposes a functional stack for PHM that can become a reference architecture for developing or purchasing a PHM system for an organization. The stack of PHM services ranges from the data acquisition layer through analysis functions to supply chain decision support services.

1. INTRODUCTION

There are numerous treatments of the structure of systems that are designed to provide prognostics and health maintenance (PHM) (examples of these systems is described in Section 2). They all deal with sampling data at some rate and analyzing it for some characteristic that indicates that there is a pending system or component failure. Sometimes the designs are focused on a particular aspect of PHM, e.g. a particular type of data analysis, but they all begin to take on similar structures. In addition, regardless of the system, the common goal of the various functions in PHM is to improve lifecycle costs in the supply chain or alternatively, to provide readiness and availability of components in the supply chain. In this effort, it is the business case analysis (BCA) that determines the effectiveness of PHM system functions. The BCA provides requirements for the design of PHM system functions.

A stack architecture would stratify these PHM functions into domains that are orthogonal in their system responsibilities. That is, they involve disjoint sets of activities that produce data that is consumed by the function above it. In this regard, the activities in each layer are opaque to the other layers. This paper presents a stack of functions that can be used as a reference stack architecture for PHM. The reference architecture is driven by the analysis of various existing PHM architectures and the extraction of a commonality from them.

Stacks of functions or responsibilities are used in systems architecture to partition the subsequent design activities and enable reuse of functions. They enable the development of clean interfaces between system functions in that a layer only consumes the information from adjacent layers. The reference stack presented here can enhance the implementation of the BCA requirements because it exposes PHM functions in a way that makes them transparent to PHM system architecture development. There is less of a chance that the PHM system would incur an unforeseen cost due to unnecessary development. Supply chain stakeholders can agree on the functional structure of the system and understand the level of effort that is required to develop the system.

Section 2 surveys a few PHM systems that have been described in the literature in order to extract some common functionality for the discussion in Section 3 that organizes PHM functions into the reference stack.

Clearly there are far more systems than are discussed in Section 2, but the structures are similar. The intent is to develop a motivation for the common stack of system functions in Section 3. For detailed descriptions of these systems and others, the reader is referred to the references and literature. A goal of this paper is to be able to employ its results in evaluating as well as designing PHM systems.

Charles Crabb. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND

This section reviews some previous PHM architectures in order to extract some commonality to support the discussion of the reference stack architecture that is appears in Section 3. There have been several approaches to developing PHM systems. They all involve the collection of data (generally data from sensors) from a platform system that is managed, along with its assembled components, by the supply chain. The collected data is analyzed by either a data driven approach, which performs statistical analysis of the data or a model driven approach, which develops a physical model in order to trace the behavior of the data to specific components on the platform (Analysis is discussed in Section 3.2). Both techniques can be employed in an analysis. The results of the analysis are then transmitted to consumers such as the decision process in the supply chain. The results of analysis support maintenance and component management in the supply chain.

Overall, the results of PHM produce several benefits to the supply chain: there is a cost and availability benefit to lifecycle systems management that is driven by the acquired capability to defer maintenance optimally and therefore lower maintenance costs. In addition there are benefits that are orthogonal to life cycle cost such as improved component and system reliability and safety.

The following sections review some of these PHM architectures.

2.1. PHM Architecture Driven by a Systems Engineering Approach

Begin (2012) describes a general architecture that is derived by applying systems engineering principles to PHM. This work develops a methodology for producing a "solution-neutral" PHM architecture. A functional decomposition is given in Figure 1. The various components that are associated with PHM are given but there is no connectivity to the supply chain decision-making services to complete the requirements of the supply chain for lifecycle cost management.

Nevertheless, the simple functional structure in Figure 1 forms a basis for formally defining what a PHM system is. Data acquisition is fundamental and provides, generally, time series data that the other layers consume. Diagnostics obviously looks for failed or faulted components while the more difficult to achieve prognostics might sit on top of diagnostics and use diagnostic services to produce a prediction of remaining useful life of a component. Finally, health management provides overall condition maintenance data to the supply chain, which is not shown. Testability is a function that can be a logical activity that is absorbed into each of the functions.

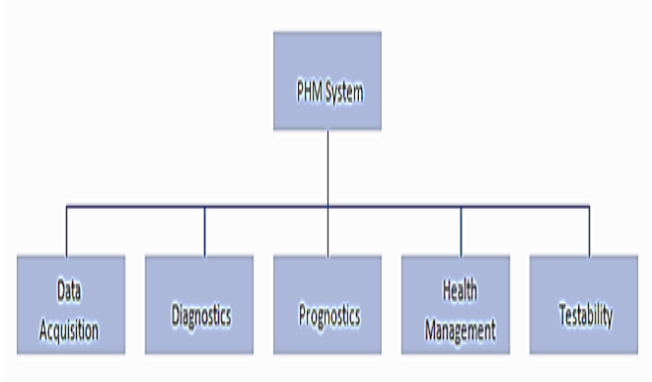


Figure 1. Notional PHM architecture (Begin, 2012).

2.2. Boeing IHVM Reference Architecture

Boeing developed a comprehensive integrated vehicle health management reference architecture (Keller, Wiegand, Swearingen, Reisig, Black, Gillis & Vandernoot, 2001) that is shown in Figure 2.

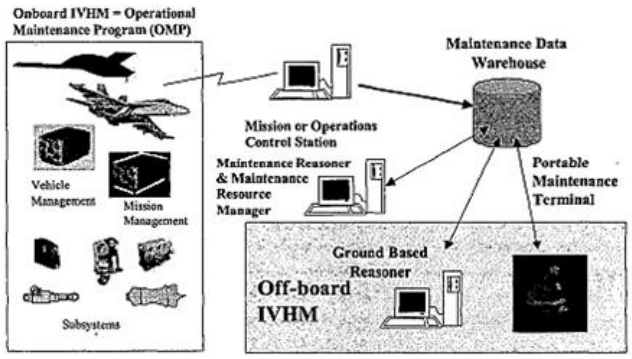


Figure 2. The Boeing integrated vehicle health management architecture (Keller, et. al. (2001)).

PHM functions are distributed from the vehicle to analysis activities that are off-platform and a data warehouse. Their partitioning is dependent on the characteristics of the infrastructure such as network bandwidth.

A reference stack of PHM functions is given in Figure 3 where PHM data flow from sensors at the bottom through signal processing of the data, monitoring of component condition, developing a health assessment of the platform and then a prognostic estimate of component lifetime which is a remaining useful component life. Decision Support is the recipient of the analysis results that uses them for lifecycle management in the supply chain. The presentation layer represents peer-to-peer communication with stakeholders in the supply chain.

All of the PHM functions are present in the stack in Figure 3 and the reference stack that is developed in this paper is similar to it and will be discussed in Section 3.

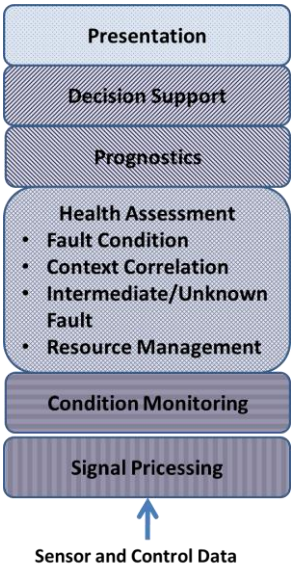


Figure 3. Stack of functions in the Boeing reference IHVM architecture (Keller, et. al., 2001).

2.3. Distributed Prognostic System Architecture

Expanding the diagrams in Figure 1 and Figure 2, the prognostic results of the PHM analysis function can be transmitted to the decision support function via publish and subscribe services. In the architecture in Roemer, Byington, Kacprzynski and Vachtsevanos (2006) that is shown in Figure 4, data flows to logistics decision support at the top. Analysis algorithms are specified to be at the lower level in the stack that is located at the subsystem that is under observation. A reasoner hierarchy isolates fault regions with reasoners placed at the subsystem in addition to the platform so the analysis function can be distributed vertically in the stack.

The functions in Figure 4 begin to look layered and the publish-subscribe mechanism with a data pipe for big data defines clean interfaces that support both the sharing of data and the opacity of the functions that produce the data. Section 3 will organize these functions into a reference stack of PHM architectural functions.

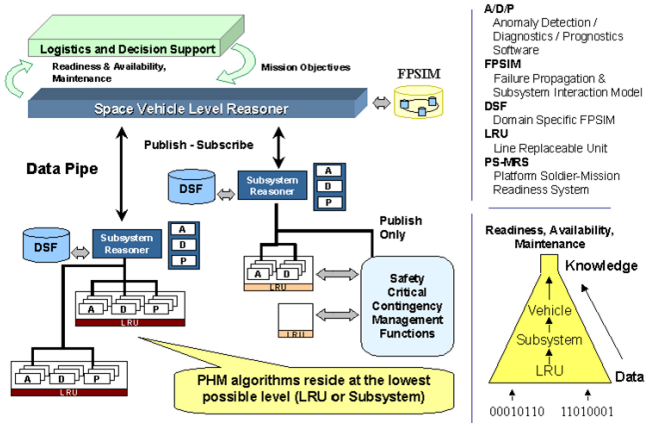


Figure 4. The distributed prognostic system architecture in Roemer, et. al. (2006).

2.4. Distributed PHM Algorithms

Saha, Shaha and Groebe (2009) give a more tightly coupled distributed PHM architecture. In this architecture, shown in Figure 5, the analysis functions are distributed onto sensors with computational elements (CE). The Central Server assists in analysis if there is insufficient remote computational capability, for example in the particle filter algorithm that creates the RUL probability distribution.

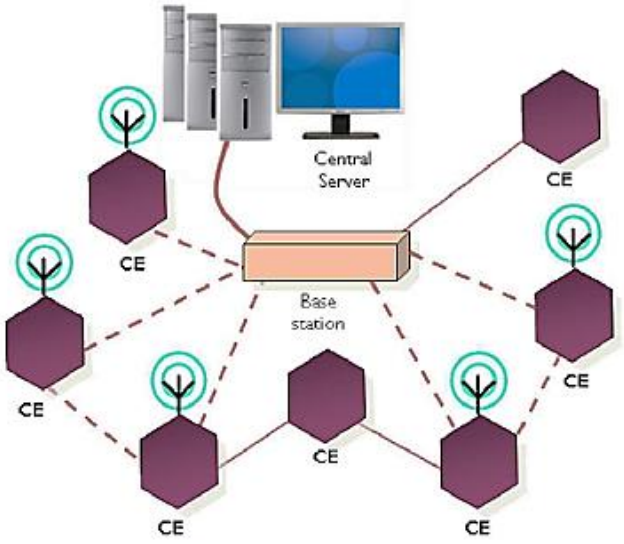


Figure 5. A distributed PHM architecture of sensors with computational elements that perform analysis (Saha, Saha & Groebe. 2009).

Were this architecture to be fit into a stack structure, the analysis function is still layered above data acquisition even though it is distributed. As will be seen in Section 3 the stack is devoid of deployment strategy because it is a logical functional specification.

Again, absent in this architecture is the connection to the supply chain support services in the enterprise that is in Figure 4. In developing a BCA for such an architecture, the supply chain connectivity needs to be considered because it is the recipient of the benefit. Section 3.3 discusses the value of this connection in relation to the BCA. Clearly, this system could be integrated into the supply chain with a services interface because of its computational capability. The connectivity at the level of the CE would be less complex (have a simpler data transfer) due to limitations to computation.

In organizing this architecture in a functional stack, the communications functions to/from the CE can be specified as another interface to the analysis function from the data acquisition function. This will be developed in Section 3.

2.5. F/A-18 Inflight Engine Condition Monitoring Architecture

Hall, Leary, Lapierre, Hess and Bladen (2001) present a PHM architecture for the F/A-18, known as the Inflight Engine Condition Monitoring System (IECMS), shown in Figure 6. The IECMS is an end-to-end system in which the sensor data is retrieved from components on the aircraft in the upper left of Figure 6 and transferred to analysis functions at the ground station that provide results to the pilot, maintainer and maintenance control on the right.

The data collection architecture on board the aircraft is further specified, producing a stack of responsibilities from the component that is sensed to the maintenance stakeholders. The functions in this design also support the reference architecture that is developed in Section 3.

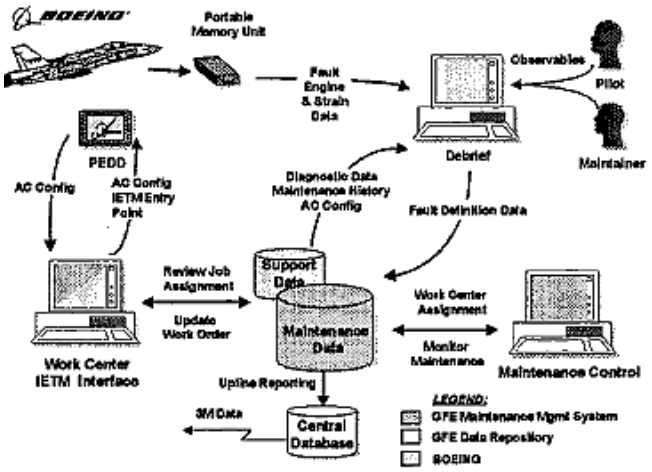


Figure 6. The F/A-18 Aviation Maintenance Environment (AME) Ground Station (Hall, Leary, Lapierre, Hess, & Bladen, 2001).

In Figure 6 the data store is the central part of the architecture. The data types are described in the data transfers between the nodes, for example the fault data from the aircraft and the maintenance data from the data store. The pilot and maintainer on the right form the decision support services (at least part of them) in the supply chain. This architecture conforms to the stack in Figure 3.

2.6. PHM Architecture for Defense

Butcher (2000) presents an architecture that describes a condition-based maintenance system for the Department of Defense. It functionally decomposes into many of the components that are in the architectures that have been presented so far. Figure 7 shows the architecture diagram.

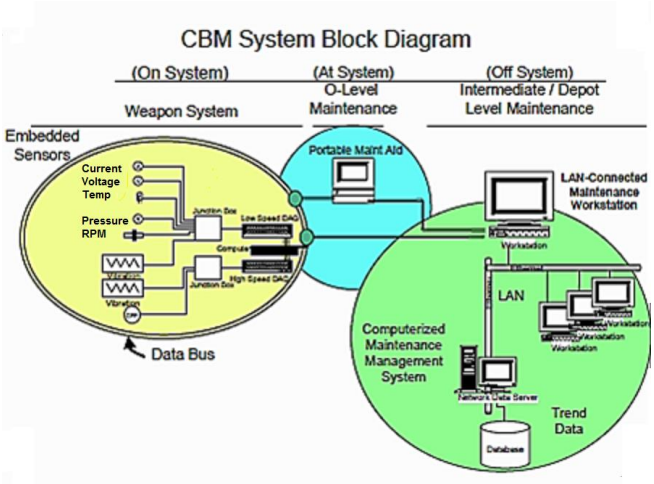


Figure 7. The condition-based maintenance architecture for the DoD in Butcher (2000).

The architecture in Figure 7 defines three domains, On System, At System and Off System. The functions in the stack in Figure 3, Sensor and Control Data and Signal Processing are located on-system. The At-System domain is meant to be the data collection function with the Portable Maintenance Aid shown in the figure. In the operational environment, connectivity can be reduced, hence there is reliance on a physical means of data transfer off the platform; computing capabilities on platform can be reduced hence analysis functions are moved to the right. However, Butcher does discuss the richer on-system computing environment of the Joint Strike Fighter that is diagrammed here in Section 2.8 and conforms to this architecture. Supply chain decision support services are in the Off System domain on the right with the data store. This diagram conforms to the stack of functions in Figure 3.

A similar architecture is developed in Section 3.3.2.

2.7. Condition Monitoring Framework

Another way to partition a PHM architecture is by its ontological elements. There is a description and data hierarchy of what PHM functions do, and Emmanouilidis,

Fumagalli, Jantunen, Pistofidis, Macchi and Garetti (2010) develop an architecture by including the knowledge base involved with condition-based monitoring, notably, "Physical Assets, Networking, Knowledge Management, Computational Models and Usage of Information & Operational Technology". The result is Table 1.

Table 1. A condition monitoring relational data table (Emmanouilidis, et.al., 2013).

Physical Assets	Networking	IT/OT	Maintenance Knowledge	Computational Model
System	MAN/WMAN, LAN/WLAN, 3G/4G	ERP, Servers	System class	State
Sub-system	LAN/WLAN	ERP, MES, CMMS, SFCS, Desktop/Server	Sub-system class	Sub-system-level Novelty Detection Diagnostics Prognostics
Unit	LAN/WLAN PAN/WPAN Gateways	Sensors, Actuators, Controllers, DAQ, RFID, PDA	Unit class Unit-level Fault modes Fault mechanisms Fault severity Fault criticality Asset relations Fault symptoms Fault features Measurement characteristics	Collective Models, Single Node Models Unit-level Novelty Detection Diagnostics Fault modelling Prognostics
Component	Serial/Bus PAN/WPAN	Sensors, Actuators, Controllers, DAQ, RFID, PDA	Component class Fault modes Fault mechanisms Fault severity Fault criticality Asset relations Fault symptoms Fault features Measurement characteristics	Single Node Models Novelty Detection Diagnostics Fault modelling Prognostics

In Table 1 there is a stack of physical assets in the left column, but these are quite different from the PHM assets that were described in the architectures in the previous sections, notably the stack in Figure 3.

The functions across the top of Table 1look like they could be organized into a stack that is similar to the one in Figure 3, but there are knowledge areas and models. What is useful in Table 1 is the compilation of the semantic terms for condition-based maintenance. The Maintenance Knowledge column organizes the fault information, which is critical to condition-based maintenance. The Computational Model column has analysis areas. The networking column has the connectivity units. It is as if this architecture could reside on top of the other architectures that are described in this section. As such it is a *semantic* architecture that could reside in a communications stack.

This paper develops a stack of PHM functions in Section 3. It is along the lines of the stack in Figure 3. However, the knowledge base needs to be developed for web services that communicate the PHM analysis results throughout the supply chain. Thus, building a table such as Table 1 creates the PHM ontology for an enterprise. In the stack that is introduced in Section 3, it is viewed that the organization of terms in Table 1 is assembled at the enterprise level where those terms have meaning across the entire PHM system area of operation. Ontologies are discussed later on in Section 3.3.2.

2.8. JSF Autonomic Logistics Architecture

The F-35 has the most recent and complete PHM system for a complex operating environment, having to monitor the F-35 and its F135 engine. The system detects faults and predicts component lifetimes, the results of which are transmitted to the maintenance activities on the ground where aircraft components can be managed autonomously. That is, some maintenance activities are replaced by the PHM results.

The development of prognostic functions is ongoing, but the supply chain is able to respond more quickly to aircraft maintenance needs than was previously possible (McCollom & Brown, 2011). There is a concept of operations for delivering analyzed data autonomously from the aircraft to the supply chain in order to reduce supply chain inefficiencies. There is a stack of responsibilities from data collection at the aircraft to the decision support functions in the supply chain.

The layers of the architecture are shown left to right in Figure 8. Again, each of the areas of responsibility can be broken down into detailed stacks of functionality. The Air Vehicle has analysis and fault detection functionality and a function that manages that data. Fault data from the aircraft is then reported directly to the Decision Support Maintenance Planning Condition-based Maintenance node on the right, in the spirit of the stack in Figure 3.

The following section suggests a structure that generalizes all the architectures that were discussed in this section.

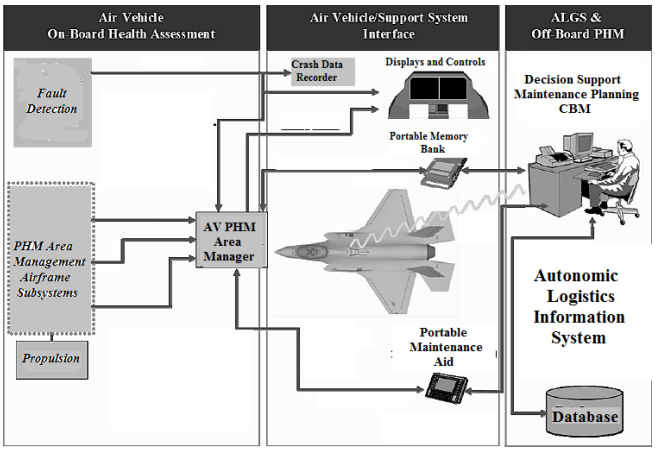


Figure 8. F-35 Autonomic logistics system deployment (McCollom & Brown, 2011).

3. THE PHM REFERENCE STACK

This section synthesizes the discussions of the architectures in Section 2 into a general reference stack of PHM functions.

The development of PHM architecture is based on requirements for integrating PHM into the supply chain. The requirements are generated by a business case analysis (BCA) that justifies the cost of developing a PHM system against the cost of managing the traditional supply chain for a particular system (Beyer, Hess & Fila, 2001, OSD(ATL), 2010).

From the discussion in Section 2, functional layers can be identified that reflect the activities in the various architectures that meet some need for PHM in a supply chain. The stack in Figure 9 is an architectural response to the requirements for PHM and defines the functions that produce the required return-on-investment (ROI) or increased availability. The BCA-generated requirements are the input on the left.

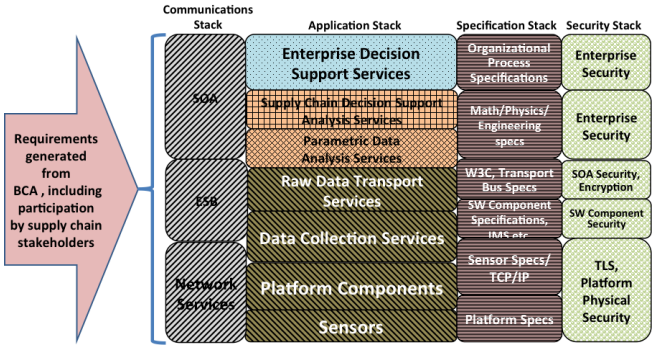


Figure 9. Functional reference stack of PHM Services.

Figure 9 organizes the architectural functions into function-type areas in the columns. In the beginning, a set of requirements is generated from a process that builds a business case analysis for the system, shown in the arrow on the left. Such a process includes interaction with suppliers, integrators and the other stakeholders in the supply chain. This process is beyond the scope of this paper, but provides the motivation and formal requirements for developing the system that delivers PHM functionality.

The central area is the Application Stack that delivers the system functionality. Other stacks support the Application Stack: A communications structure is on the left. The Specifications Stack provides technical requirements and standards that are levied on the system. The Security Stack on the right satisfies the information assurance requirements of confidentiality, availability, integrity, auditing and so forth.

In discussing the architecture in Figure 9, it is best to start at the top, because the motivation for developing a PHM system is to improve the cost of managing the supply chain that supports systems in use by an organization. Every logistics organization has Enterprise Decision Support Services at the top layer where supply chain activities manage parts and services for the systems that an organization deploys. The bottom layers produce the information that enhances supply chain activities.

The following sections discuss the elements of the stack in more detail.

3.1. Enterprise Decision Support Services

The decision support services in the enterprises are the ultimate recipient of PHM data. As mentioned above, the goal is to create a greater efficiency in the supply chain that improves its operating costs by streamlining the management of the systems that are under its control. Decision Support that consumes the products of the Analysis Services that are located beneath is shown at the top in Figure 9.

Following the information flows from the architectures that were described in Section 2, Figure 10 abstracts the flow of the PHM analysis products to the Supply Chain Customers that are in the upper right who receive analysis results from the PHM functions that are in the bounded region.

The Data Collection Services in Figure 9 can produce a large amount of data, such as time-series data, to be analyzed. The analysis *results* are greatly distilled from the raw, parametric, data that is produced by the Data Collection Platforms that are shown on the platforms within the bounded region in Figure 10. For Data Collection–Platform 1, mid-left in Figure 10, the analysis function is actually on the platform and analysis results are pushed up into the enterprise, as in Figure 4 above.

It is important to note that Figure 9 is a logical structure; the functions are logical and are deployed in the implementation phase, thus can be located where the system design dictates. For example, the architectures of Roemer, et. al. (2006) in Figure 4 and Saha, et. al. in Figure 5 distribute Analysis Services to remote elements in the design.

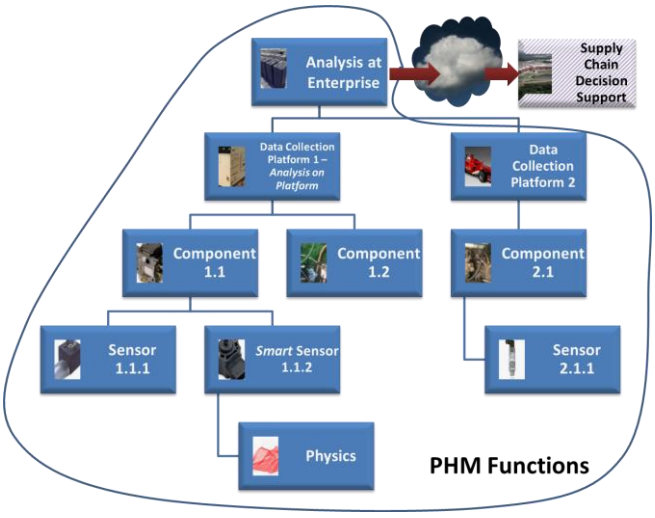


Figure 10. Injection of PHM analysis results from two platforms into the supply chain.

The analysis results that are pushed to the supply chain, or, to which the enterprise subscribes in the case of a cloud-based Services Oriented Architecture (SOA), can have some governed ontology that deals with components, fault modes and prognostics such as that described by MIMOSA (2009) and ISO 13374-3:2012 (2012). Section 2.6 described a semantic architecture, and the use of a SOA is further discussed in Section 3.3.2.

MIMOSA is a stack-oriented data architecture. Figure 11 shows its stack of functions, starting from a layer that deals with the acquisition of data, through layers that further refine the data. Analysis occurs in the HA and PA layers, the results of which generate an advisory in the AG layer.

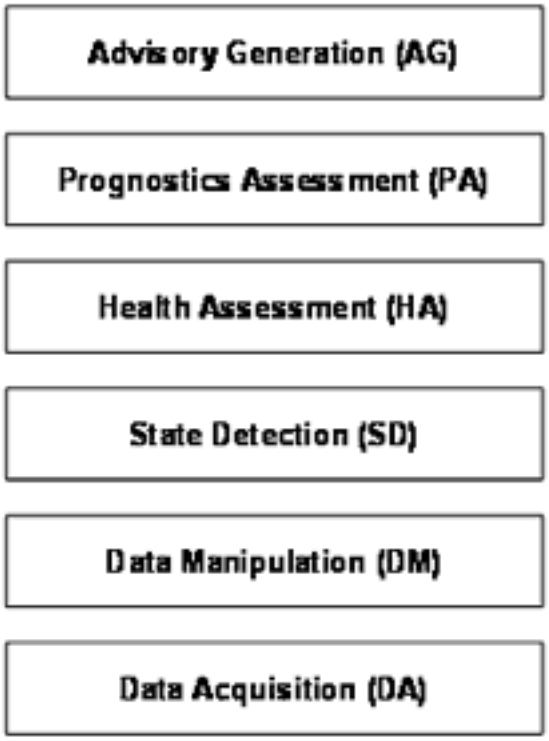


Figure 11. OSA CBM functional blocks (ISO 13374-3:2012).

Figure 11 is suggestive of functions in the PHM stack that is shown in Figure 9, and it is used in a *data architecture* that defines the interfaces between these PHM functions. The functions in in Figure 9 are *system* functions. The data that is generated in its layers can conform to the stack that is shown in Figure 11 but the functional organization of a PHM system is not mandated to conform to the functional organization that is shown in Figure 11.

An advantage of the ISO 13374 standard is that a schema can be built from its data stack. This is described in Section 3.3.1 for tagging the data and in Section 3.3.2 for developing an ontology for PHM web services.

Figure 10 indicates that there might be some sort of service-level agreement (SLA) between the supply chain decision support and the analysis services and governance over the analysis results that are created and its users if data is shared over a SOA. There is a cost to providing the SOA transport. Design costs as well as operating expenses of the SOA need to be factored into the BCA. The SLA includes requirements for quality of service (QOS) which involves the required bandwidth for data transfer. In Figure 4 above, a data pipe is inserted for a higher level of performance for more massive data amounts such as time-series data. Section 3.3.1 discusses the QOS requirement for raw data transport.

There is a central notion to this paper that the creation of a PHM system should not require the refactoring of supply chain functions in a dramatic way, as this would lead to additional cost. Most supply chains today operate within some sort of enterprise resource planning framework that is connected via a SOA. Thus, it is convenient to publish the PHM analysis results without having to retool the data transport mechanism. Section 3.3.2 discusses the SOA in stack.

The functions in Figure 9 and Figure 10 are to be used in the complete deployment of PHM technology to the platform and can used in a BCA to develop the cost basis for the PHM system (Sandborn & Wilkinson, 2007, and Kent & Murphy, 2000)).

3.2. Analysis Services

The Analysis services process the raw, parametric, data received from the Data Collection layer that collects data from the critical components that were identified in the BCA and provide supply chain decision support for life cycle management of components. The following two sections discuss these modes of analysis.

Analysis activities can be broken out into a stack of functions once the target components are identified. A good discussion of the analysis process is given by Roemer, et. al. (2006) and there are numerous approaches.

3.2.1. Parametric Data Analysis

The result of the analysis process in PHM is generally some sort of estimation of remaining useful life of the components from a trend in the data that indicates the *future* behavior of the component. The well-known idea is to predict a remaining useful life of a component, among other analysis products, that is injected into Supply Chain Decision Support to expedite its product stockage and provisioning functions, as is shown in Figure 10.

The notion really is that there is some stochastic process in state space which is therefore non-deterministic, but were the process known, would show a path to failure where the operation of a component passes into a region of

inoperability. The time from the detection of this path, t_c, say, to the time of failure, t_f, is the component's remaining useful life or RUL. Figure 12 illustrates the path in state space of some measured parameters from a component. Multiple parameters are more difficult to correlate, so generally, papers on RUL deal with only one parameter. Figure 12 reminds us of the underlying physical complexities of the problem by plotting a multi-dimensional state space.

The broadened paths indicate that the parameters are really described by a probability density function with some statistical moment, such as average, indicated by the narrow path curve. Thus, the remaining useful life calculation is some probability distribution function. Examples of stochastic treatments of RUL are in Saha, Goebel, Poll and Christophersen (2007), Tang, Kacprzynski, Goebel and Vachtsevanos (2009) and Sankararaman and Goebel (2013).

If we *knew* the entire path for all time, we would see that somewhere along the permissible operating range the path bifurcates into a course that leads it to failure at some point in the future. In Figure 12 that point is the red statistical ball, the "failure occurrence volume", where the failure curve penetrates the operating volume at time t_f.

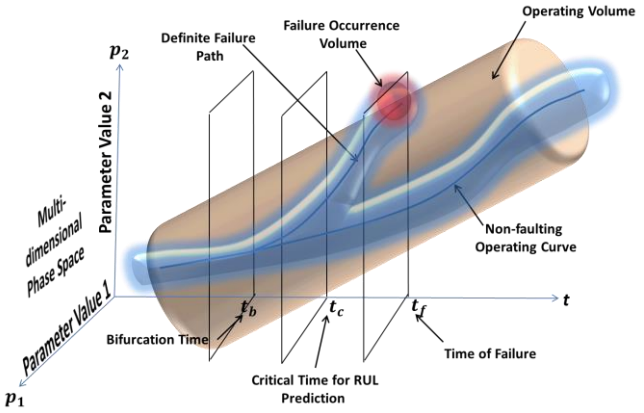


Figure 12. Paths, operating and failure, of measured component parameters in phase space.

The program in PHM is to recognize that the system is operating on this failure path soon enough to get the information back into Supply Chain Decision Support so that it has time to provision stockage and provide maintenance and part management functions before the actual failure, as is well known. In Figure 12 the time t_c is the time before the failure time t_f that the prediction of failure occurs, and the knowledge of which enables the supply chain to act on the predicted failure cost-effectively.

Analysis services that produce prediction of failure, or remaining useful life incur cost in the BCA. There is an open ended-ness to the analysis process because new failure modes or behavior characteristics can be discovered by continued analysis, but this is difficult to budget in a BCA. The identification of *specific* analysis algorithms that are attached to specific failures enables a turn-key system. Such algorithms would be deployed in distributed systems such as that is shown in Figure 4 and Figure 10 where analysis is distributed to the sensor locations. It may be necessary to update the remote CEs in Figure 4 with new algorithms (updating algorithms in regards to the stack in Figure 9 is discussed in Section 3.6). The identified analysis algorithms could conform to standard measures that are implemented in libraries in analysis tools. Standards are discussed more in Section 3.5.

In regards to the stack in Figure 9, the interfaces to the Analysis layer need to be defined so that they are useful to the subscribers in the Enterprise Decision Support Services above. Clearly, the schematic analysis result in Figure 12 is *opaque* to the enterprise that is looking for extracted information from it such as RUL. The RUL is to be injected into the level of analysis that is at the enterprise that requires a greatly reduced and far more descriptive set of data than is raw data. Section 3.6 discusses these analysis results in relation to the orthogonality of the functional layers in the stack and the next section discusses analysis at the enterprise.

Another aspect of Analysis Services is identifying an approach to the analysis. As was mentioned in Section 2, analysis methodologies involve data-driven and/or model driven analysis processes (Bernstein, Hauske & Hermann, 2014, Byington, Roemer & Galie, 2002). In including the analysis strategy in the Analysis Services in Figure 9, it should be understood that the analysis methodology can impact the supply chain and the BCA because there can be added cost to developing a dynamical physical behavioral model for a component which is a requirement of the model driven approach.

Detecting the data signature of a failure mode and associating it with a component is generally done with a model because it is difficult to run a statistically significant number of components to produce a failure signature for the data-driven approach and run the components to failure. Therefore, a model can simulate the data that is produced by deployed sensors and trace it to a fault condition.

However, the data-driven approach can *also* incur inordinate costs if a seeded-fault approach (Hess, A. (2002)) is used to identify failure signatures because system run time is required to associate the faults with data signatures. A problem with data-driven analysis is the uncertainty of achieving a logical connection between the analysis results and the physics at the data collection point.

In reality, there is a model that is developed from behavioral equations that describe the physical behavior that produces the sensor data. This is a complex, boundary value problem to develop and solve. Therefore, a data-driven approach appears attractive, but the sensors were applied to the

critical component with some understanding of the underlying physical behavior of the system which causes the component to fault. To resolve these issues, the discussion of how the analysis process of a proposed PHM system affects the BCA for the system needs to occur with the engineering community as well as supply chain managers.

3.2.2. Supply Chain Decision Support Analysis

In Supply Chain Decision Support Services (DSS), another level of analysis occurs to determine the best plan of action for managing a component's total life cycle given the results from the PHM analysis in the previous section, as is shown in Figure 9. Thus, in developing a BCA for PHM, the Supply Chain analysis activities need to be taken into account in order to estimate the impact that PHM has on component life cycle management in the supply chain.

Such a supply chain analysis model is done in Feldman, Jazouli, and Sandborn (2009) who consider the costs of integrating PHM into the supply chain in their stochastic model for a Boeing 737 display, and Banks and Merenich (2007) develop a trade-space tool that calculates the cost benefit analysis for a PHM system for batteries in vehicle power systems.

Tsoutis (2003) simulates the effect of the autonomic logistics system for the F-35 that is shown in Figure 8 on supply chain management. He incorporates existing maintenance data for the F/A-18E/F F-414 engine (F-35 maintenance data was of course not yet available). His work compares a baseline of the traditional logistics system for the F-414 engine with a set of *modified* repair activities that are streamlined by the injection of prognostic information from a PHM system in the autonomously enabled aircraft in Figure 8. Tsoutis was able to perform a sensitivity analysis of the effects of increased component (module) reliability and prognostic accuracy, among other parameters. This type of work enhances the development of the BCA and produces a clearer understanding of the effect of introducing a new PHM system into a supply chain. The cost of restructuring a supply chain must be included in the BCA, and simulation of supply chain activities can demonstrate a cost benefit of the PHM system.

It is clear that the Analysis Services function occupies an important area for the BCA. The stack in Figure 9 can be used to partition the types of data, components and types of analysis to determine how much effort is required to reach a result from the analysis function.

3.3. Logistics Data Transport

The transport mechanisms for PHM data are shown in Figure 9 on the left as the Communications Stack and below the Analysis layer as Raw Data Transport Services. Raw Data Transport Services provide primitive data ("raw data" or parametric data), principally sensor data that can be voluminous due to high sampling rates, to the services that analyze and transform it. Discussion of their mechanisms of transport is treated separately for this reason in the next section.

The general Communications Services provide higher level communications functions such as a web services stack that includes semantic information and can be governed by an ontology. They are discussed in Section 3.3.2.

3.3.1. Parametric Data Transport

A central activity in the BCA for PHM is generally based on identifying high cost components that are expensive to manage (Banks, Reichard, Hines & Brought, 2008). Failure characteristics of these components are identified through a failure modes, effects and criticality analysis (FMECA) that leads to a root-cause analysis of the failure. The process identifies a characteristic of the component that can be monitored by sensing a region on the component that produces data that identifies that characteristic (See also the discussion in Section 3.2.1). The sensed data quantity can be large due to the results and recommendations of the FMECA and root-cause analysis. As the stack in Figure 9 and flow diagram in Figure 10 show, the resulting collected data needs to be transported to the analysis services that detect the failure characteristics.

Sensors generally sit on buses such as the well-known SAE J1939 and MIL-STD-1553 buses. Determination of what bus to use is dependent on the particular connectivity with the sensor. The Data Collection Services in Figure 9 define the data collection protocols. The cost for PHM systems on new equipment or retrofits includes the technology in this layer. These standards would be included in the standards in the specification stack, Section 3.5

The communications protocols for raw, parametric, data need to provide enough quality of service (QOS) to support large data streams. The protocol for transporting the data might not be the internet, but some physical mechanism such as a local lap top computer or portable maintenance aid. This configuration eliminates the bandwidth bottleneck but lessens real time data collection. However, the Analysis Services could be deployed on the local laptop. Thus, the stack in Figure 9 is useful for partitioning the functions of the PHM to the deployed areas in the system. Figure 13 shows a possible deployment. QOS in regards to data transport to the enterprise is discussed in Section 3.1.

Using the Stack to Reduce Bandwidth Requirements

Sensor data is generally a time series that is obtained at regular intervals at a specified sampling rate. The physics of the problem determines what rate is required to discover the data signature that indicates pending failure, as was discussed in Section 3.2.1. As such, the data can be large and hence require high bandwidths. The cost of managing big data needs to be integrated into the BCA for the PHM system. The stack in Figure 9 is helpful because it can be

used in conjunction with the *deployment* of PHM services in the supply chain, such as that shown in Figure 10 and those that were discussed in Section 2.. A tradeoff analysis of the location of Analysis Services can reduce the cost of bandwidth. For example, Figure 5 distributes the analysis function to local computational elements that is a measure that greatly reduces the burden of having to supply a high-bandwidth transport for sensor data.

Describing or Tagging Sensor Data

An ideal is to publish this raw sensor data in the context of a services-oriented architecture. Sensor data can then flow up the left SOA stack in Figure 9.

There are formats that tag sensor data in order to develop a publish/subscribe mechanism at the parametric data level. This adds overhead to the sensor data but for large data transfers the headers are relatively small. One well known format is in the NASA CDF applications library (CDF User's Guide, 2012). A data tagging standard was built on top of that for PHM systems by the US Army known as Army CBM Bulk Data (ABCD) format (US Army PEWG, 2011). The tagging in ABCD format respects the data layers that are found in the MIMOSA standard (MIMOSA, 2009) and in ISO 13374-3:2012 (2012).

All these standardization activities need to be included in developing the BCA for PHM. The advantage is that developed standards such as NASA CDF come with functional software applications programmer interfaces that eliminate the cost of new software development. The Standards Stack in Figure 9 is useful to organize the standards at each level of the architecture. Standards are discussed in Section 3.5.

3.3.2. Communications Services

Communications services transport logistics information throughout the supply chain. The stack diagram in Figure 9 and data flow diagram in Figure 10 illustrate that these are the communications services that inject derived analytical information into the supply chain.

It is also envisioned that there is a Services Oriented Architecture (SOA) to provide the transport. An enterprise service bus (ESB) (Chappell, 2004) that provides connectors and messaging services as well as other functions in the SOA stack enables the SOA.

Enterprise Services Bus

The services stack in Figure 9 is meant to be integrated into an existing SOA that is provided by the supply chain that requests PHM technology. Adding the additional cost of developing a SOA to the PHM BCA would be excessively costly. Furthermore, it is a distinct advantage to be able to publish PHM logistics data to *existing* supply chain services that already make use of enterprise services technology.

Web Services (SOA)

Web services itself provide a stack of functions (W3C, 2004), but the configuration varies widely with providers. The supply chain would have a services architecture with governance and provisioning already determined. Thus, the PHM system should be able to publish the analysis results to the supply chain that subscribes to it. Again, it is meant to require minimal effort to connect to the supply chain. Section 3.1 discussed the role of a SOA in communicating PHM analysis results.

Figure 13 illustrates a possible deployment of a SOA architecture for PHM data transport in a military environment. Here, an ESB is located at each of the nodes in the Tactical, Operational and Enterprise areas and implements a SOA stack. Data collection on the Vehicle Platform in the tactical environment involves both the SOA for analyzed data, which can be generated on-platform, and a fast pipe, such as that in Figure 4 for parametric data transport. There is data analysis on platform as in Figure 4 and Figure 5. Raw/Parametric data is transferred to the operational node via a maintenance support device, such as a laptop or PDA over the high bandwidth link to support the bandwidth requirements. Decision support services (DSS) exist at the Enterprise node where the analysis is the decision support analysis that is discussed in Section 3.2.2 for lifecycle support. Note that the functions in the stack in Figure 9 are deployed in Figure 13 and the deployment looks like that in Figure 10.

The SOA makes the sharing of data seamless, but the diagram indicates that there has to be a common understanding of terminology of data types in the supply chain. In this environment, the ISO 13374 tagging (See Section 3.3.1) is useful for sharing the raw, parametric, data.

The ontology produces a common knowledge base throughout the supply chain and between enterprise domains; the ontology can be used to communicate diagnostic, health and prognostic information from one logistics domain to the next. The data schema, such as parametric data from the Tactical node in Figure 13, is governed by the MIMOSA standard (MIMOSA, 2009) as shown in Figure 14. The enterprise is the locus of domain expertise and has a *domain-specific* ontology that is developed by stakeholders.

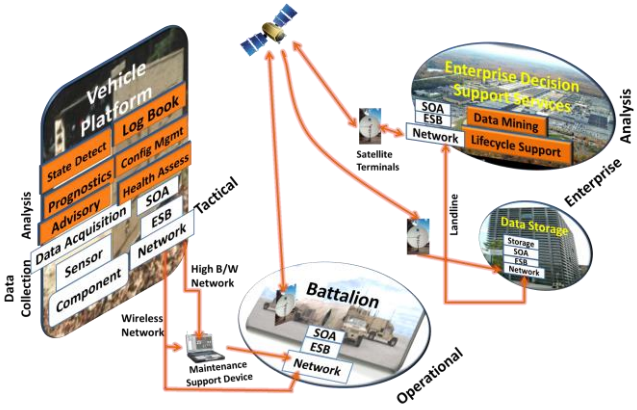


Figure 13. Notional deployment of a PHM system incorporating a SOA with corresponding ESB.

The organization might further develop an ontology to enable semantic structure to the data (W3C Semantic Web, 2014). An ontology is developed by Emmanouilidis, et. al., (2010) which was shown in Section2.6; the domain structure is shown in Figure 14. It is at this point that a table such as Table 1 can be developed at the enterprise to identify the semantic elements in the PHM system that is to be designed.

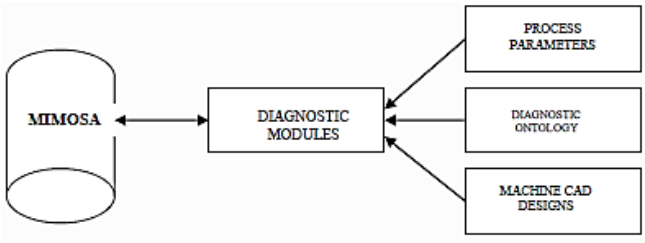


Figure 14. Diagnostic ontology of Emmanouilidis, et. al., (2010).

Including the cost and effort of developing a PHM system ontology in a BCA is a complex task. It is of course better to incorporate existing ontologies such as MIMOSA and the data stack that is described by ISO 13374. In developing a domain-specific ontology, the BCA should investigate existing ontologies in the organization and look to expand them with PHM terminology. Barring that, the effort needs to be closely monitored and costs need to be estimated as early on as possible. Ontologies are best restricted to systems of common functionality in order to bound the effort.

3.4. Security Stack

The architectures that are discussed in Section 2 are shy of data security measures, in part because PHM is the central function of the architectures and in part, because the data is not mission-critical data. For example, on an aircraft there is system control data, which is of course critical, while PHM data is produced in order to monitor the operation of the controlled system.

Data streams of time-series data such as that shown by the data pipe in Figure 4 can be protected by link encryption while services that produce higher level semantically governed information can be protected by implementing standards such as WS-Security (OASIS, 2004).

The stack on the right in Figure 9 addresses the security for the layers in the Applications Stack. What would be filled in here for the implementation are the specified security standards that are going to be used to provide information assurance to the stack. In developing the BCA for the particular PHM system that is under consideration, the cost of security may be relevant.

In developing PHM for military systems (Butcher, 2000), there are well-defined directives and procedures that need to be followed, indeed, *required* to be followed, such as *Net Ready Key Performance Parameter*, (NR KPP)CJCSI6212, 2012) and *Information Assurance Certification and Accreditation Process (DIACAP)*, (DoDI 8510.01, 2007). The implementation of DIACAP requirements requires additional time and obtaining a formal authorization to operate the system.

Collocating PHM data services with secure areas could incur a cost, possibly a cross-domain solution that would have another form of certification (DISN, 2004). The stack in Figure 9 is useful for identifying *where* data is produced in order to identify the security boundary of the originating systems (NIST 800-18, 2006), a primary task in information assurance.

3.5. Specification Stack

As mentioned at the end of Section 3.3.1 that discusses tagging sensor data by using specifications for various PHM functions, the stack in Figure 9 organizes standards and exposes them to the developer community for evaluation of their effectiveness. In Figure 9, the Specification Stack is to be augmented with the specifications that the design incorporates.

The specification stack begins at the top where decision support activities occur to support lifecycle systems management. These standards will already be in place, as PHM architecture does not refactor the supply chain architecture; it would be difficult to justify a cost for doing so. *However*, the JSF autonomic logistics architecture in Section 2.6 *does* affect some of the organizational structure of maintenance because its autonomic prognostic notification of faulty parts on the aircraft can remove a maintenance inspection step. A discussion of a simulation of this effect was given in Section 3.2.2.

The employment of standards can affect the BCA for the PHM system; the decomposition in Figure 9 can be factored according to the envisioned and simulated cost model for

the PHM system. The use of standards over custom specifications more confidently reduces the cost of the PHM system. An example is the use of standard analytical methods for analyzing sensor data (See Section 3.2). While a lot of analysis work is specific to a particular system, the BCA can require that standard analysis libraries and tools be employed to conduct the search for failure precursors in the sensor data.

3.6. Orthogonal Property of the Stack Layers

As discussed in the Introduction, the functions in the layers in Figure 9 are isolated from adjacent layers. In good systems architecture, each layer contracts with the services from the layer below it via a well-defined interface and has no knowledge the internal functionality of its neighbors. Services do not get data from the service layers above them. One could ask if a lower layer receives data from a higher layer in the case of Figure 5, where new analysis algorithms are distributed to the remote CEs (this update process was mentioned in Section 3.2.1). However, this would be a transaction *within* the *analysis* layer in the stack in Figure 9 and would not violate the read-down-only principle. Recall, the stack in Figure 9 is *not* a deployment diagram that specifies where services physically reside.

An example of functional separation between layers is the analysis layer: the Supply Chain Decision Support Analysis Services and the Parametric Data Analysis Services layers in Figure 9. Supply Chain Decision Support Analysis Services would have no access to the raw, parametric data from the Parametric Data Analysis Services below it. Instead it subscribes to an agreed-upon analysis result, possibly component RUL, through a contractual interface. It does not have the same analysis functions or purposes of analysis as are in the Parametric Data Analysis Services layer. In terms of a BCA, this means that there is no duplication of effort between the layers and there is no confusion of functions in the layers. The argument is the same for all other layers.

4. Conclusion

Figure 9 presented a reference stack of functional areas that would comprise a reference architecture for a PHM system. Its development was motivated by the analysis of several existing PHM architectures in Section 2. Closely attached to this stack is the business case analysis (BCA) that provides a motivation for developing a PHM architecture. The reference stack in Figure 9 supports the BCA by making the functional composition of the proposed PHM system transparent to the cost and availability requirements that are generated by the BCA.

Included in the stack in Figure 9 are additional stacks that track the specifications to which the PHM system is designed. They are constraints on the cost of the system. Choosing existing specifications of services can reduce the cost of the system. On the other hand, security requirements can be a burden on the cost and availability of a PHM system. The communications stack describes the data transport functions for the PHM system but is also meant to identify open-source transports and a SOA that can be integrated easily into the existing supply chain enterprise services structure.

The PHM architectural reference stack is an effective way to communicate PHM system functions to the stakeholders in the supply chain who have commissioned the PHM system to reduce the lifecycle costs of the components that they manage.

REFERENCES

Banks, J. & Merenich, J. (2007). Cost benefit analysis for asset health management technology. *Reliability and Maintainability Symposium, 2007. RAMS '07. Annual Conference*

Banks, J.C., Reichard, K.M., Hines, J.A. & Brought, M.S. (2008). Platform degrader analysis for the design and development of Vehicle Health Management Systems. *International Conference on Prognostics and Health Management, 2008.*

Begin, M. P., (2012). *Systems engineering processes for the acquisition of prognostic and health management systems*. Masters thesis. Naval Postgraduate School, Monterrey, CA., http://www.nps.edu/

Bernstein, A., Hauske, S. & Hermann, M. (2014). Decision support systems II, http://www.elml.uzh.ch/preview/fois/DSSII/en/html/index.html Univerisity of Zurich

Beyer, B., Hess, A. & Fila, L. (2001). Writing a convincing cost benefit analysis to substantiate autonomic logistics. *Aerospace Conference, 2001,* Big Sky, MT

Butcher, S. (2000). Assessment of condition-based maintenance in the department of defense. *Logistics Management Institute*

Byington, C. S., Roemer, M. J. & Galie, T. (2002). Prognostic enhancements to diagnostic systems for improved condition-based maintenance. *Aerospace Conference Proceedings, 2002.* Big Sky, MT

CDF User's Guide (2012*). CDF User's Guide Version 3.4*, February 28, 2012. Space Physics Data Facility NASA / Goddard Space Flight Center

Chappell, D. (2004). *Enterprise Service Bus*. O'Reilly Media, Inc. ©2004

CJCSI6212.01F (2012). *Net Ready Key Performance Parameter (NR KPP)*. United States Department of Defense Chairman of the Joint Chiefs of Staff

DISN (2014). *DISN Connection Process Guide, Cross Domain Solutions,* http://www.disa.mil/Services/Network-Services/Enterprise-Connections/Connection-Process-Guide/Service-Appendices/CDS. Defense Information Services Agency

DoDI 8510.01 (2007). *Information Assurance Certification and Accreditation Process (DIACAP)*. ASD(NII)/DoD CIO

Emmanouilidis, C., Fumagalli, L., Jantunen, E., Pistofidis, P., Macchi, M. & Garetti, M. (2010). Condition monitoring based on incremental learning and domain ontology for condition-based maintenance. *Proceedings of APMS 2010 International Conference on Advances in Production Management Systems*, Cernobbio, Como, Italy, 11-13.10.2010

Feldman, K., Jazouli, T. & Sandborn, P. (2009). A methodology for determining the return on investment associated with prognostics and health management. *IEEE Trans. on Reliability*, Vol. 58, (No. 2), pp. 305-316.

Hall, C. L., Leary, S., Lapierre, L., Hess, A. & Bladen, K. (2001). F/A-18E/F F414 advanced inflight engine condition monitoring system (IECMS). *Aerospace Conference, 2001*, Big Sky, MT

Hess, A. (2002). *Prognostics and Health Management: The Cornerstone of Autonomic Logistics*, Joint Strike Fighter Program Office PHM Development

ISO 13374-3:2012 (2012). *Condition monitoring and diagnostics of machines -- Data processing, communication and presentation*. International Standards Organization

Keller, K., Wiegand, D., Swearingen, K., Reisig, C., Black, S., Gillis, A. & Vandernoot, M. (2001). An architecture to implement integrated vehicle health management systems. *AUTOTESTCON Proceedings, 2001*. IEEE Systems Readiness Technology Conference

Kent, R. M., & Murphy D A. (2000). *Health monitoring system technology assessments - cost benefits analysis.* NASA / CR-2000-209848 National Aeronautics and Space Administration Langley Research Center

McCollom, N. N. & Brown, E. R. (2011). PHM on the F-35 fighter. *IEEE Conference on Prognostics and Health Management (PHM), 2011*

MIMOSA (2009). *Common Relational Information Schema (CRIS)* Version 3.2.2 Specification, Production Release, December 31. Machinery Information Management Open Systems Alliance

NIST 800-18 (2006). *NIST Special Publication 800-18 Revision 1 Guide for Developing Security Plans for Federal Information Systems.* Computer Security Division Information Technology Laboratory National Institute of Standards and Technology

OASIS (2004). *Web Services Security: SOAP Message Security 1.1* (WS-Security 2004). OASIS Open 2002-2006

OSD(ATL) (2010). *Information on Conducting Business Case Analyses For Condition Based Maintenance Plus (CBM+) Initiative*. Report of the Office of the Secretary of Defense CBM+ Action Group 2010 Summer Study

Roemer, M. J., Byington, C. S., Kacprzynski, G. J., & Vachtsevanos, G. (2006). An overview of selected prognostic technologies with reference to an integrated phm architecture. *Proceedings of GT2006 ASME Turbo Expo 2006: Power for Land, Sea, and Air* May 8-11, 2006, Barcelona, Spain

Saha,B, Goebel,K., Poll, S. & Christophersen, J. (2007). A bayesian framework for remaining useful life estimation. *Association for the Advancement of Artificial Intelligence*

Saha, B., Saha, S. and Goebe, K. (2009). A distributed prognostic health management architecture. *Proceedings of the Society for Society for Machinery Failure Prevention Technology*

Sandborn, P. A, Wilkinson, C. (2007). A maintenance planning and business case development model for the application of prognostics and health management (phm) to electronic system. *Microelectronics Reliability*, Vol. 47, (No. 12), 1889-1901

Sankararaman, S. & Goebel, K. (2013). Why is the remaining useful life prediction uncertain? *Annual Conference of the Prognostics and Health Management Society 2013*

Tang, L., Kacprzynski, G., Goebel, K., & Vachtsevanos, G. (2009). Methodologies for uncertainty management in prognostics. *Aerospace Conference, 2009*, Big Sky, MT

Tsoutis, A. (2003). *Simulation of the I3 to D Repair Process and Sparing of the F414-GE-400 Jet Aircraft Engine*. Masters thesis. Naval Postgraduate School, Monterrey, CA.

US Army PEWG (2011). *Army Product Data & Engineering Working Group (PEWG) Report: Standards Used to Acquire Product Data Summary Report*. US Army Materiel Command

W3C (2014). http://www.w3.org/standards/ World Wide Web Consortium

W3C Semantic Web (2014). http://www.w3.org/standards/semanticweb/ World Wide Web Consortium

BIOGRAPHIES

Charles F. Crabb is principle analyst at LogTech, LLC in Wall, NJ. He received his BS in Physics from the University of North Carolina at Greensboro where his research led to an NSF grant to acquire a 36" research telescope and observatory. He received his MS astronomy from Boston University with work in ionospheric physics that modeled degradation of the ionosphere due to space shuttle engine burns. He has had an extensive career in hardware and software engineering, scientific computing and systems engineering, having worked on two new instruction sets, VLSI, computer-aided design code, information assurance, cryptographic systems, mathematical development of solids modeling and massively parallel high-energy density codes. He currently

leads a group that is focused on the analysis of enhanced sensor data to determine statistically and physically based causes of component failure in order to develop prognostic assessments.

Managing Fleet Wide Sensory Data: Lessons Learned in Dealing with Volume, Velocity, Variety, Veracity, Value and Visibility

Preston Johnson[1]

[1]National Instruments, *Austin, Texas, 78759, USA*
preston.johnson@ni.com

ABSTRACT

More than ever, asset operators and OEMs are investing in fleetwide monitoring systems. With the roll out of these monitoring systems, huge amounts of sensory data are generated. In a single Gigawatt power plant, asset monitoring systems sort through terabytes of sensory data per week. To contend with the volume and velocity of sensory data, analytics and data management techniques are employed along the life of sensory data from digitization at the asset, to storage in the information technology infrastructure. This paper presents techniques, both promising and fielded, for analytics to manage the volume, velocity, veracity, variety, and value of fleetwide asset monitoring data yielding opportunities for advanced visibility of actionable information.

1. INTRODUCTION

In industrial asset monitoring applications, scientists, engineers, and asset maintainers can collect vast amounts of data every second of every day. Drawing accurate and meaningful conclusions from such a large amount of data is a growing problem, and the term "Big Data" describes this phenomenon. Big Data brings new challenges to prognostics applications in the form of analysis techniques, search and retrieval, data integration or fusion, reporting, and system maintenance (Johnson & Farrell, 2011). All these challenges must be met to keep pace with the experimental growth of asset related data.

Take for example, the Large Hadron Collider at the European Organization for Nuclear Research (CERN), where for every experiment the control and monitoring systems can generate 40 terabytes of data (Bradicich & Orci, 2012), (Losito 2011). In Aerospace, for every 30 minutes a jet engine runs, upwards of 10 terabytes of operational data is generated. In a single journey across the Atlantic Ocean, a four-engine jumbo jet can create 640 terabytes of data. Multiply the single flight by 25,000 flights per day, and we yield an enormous amount of data (Gantz & Reinsel, 2011). This is "Big Data".

Preston Johnson. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. HISTORY OF BIG DATA

The technology research firm International Data Corporation (IDC) recently performed a study on digital data, including measurement files (think time waveform recordings), video (think thermal images), music (think ultrasonic), work order reports, and so on. The study estimates that the amount of data available is doubling every two years. In 2011 alone, 1.8 zettabytes (1E21 bytes) of data were created (Hadhazy, 2012), Figure 1. While, our (as in the PHM community) asset monitoring systems may not produce quite this amount of data, just consider the size of the data files we collect from diagnostic visits to our assets. Next consider the impact that low cost automatic data collection systems and sensors can and are having in our ability to continuously monitor and record data from our assets. Even within PHM asset monitoring and prognostics functions, the trends are similar: the amount of data available for predictive analytics is doubling every two years.

Figure 1. Data is collected at a rate that approximately parallels Moore's law.

The fact that the volume of data is doubling every two years mimics one of the electronics' most famous laws: Moore's law. In 1965, Gordon Moore stated that the number of

transistors on an integrated circuit doubled approximately every two years and he expected the trend to continue "for at least 10 years". Forty-five years later, Moore's law still influences many aspects of Information Technology (IT) and electronics. Consider that in 1995, 20 petabytes of total hard drive space was manufactured. Today, Google processes more than 24 petabytes of information every single day. Similarly, the cost of storage space for all this data has decreased exponentially from $228/GB in 1998 to $0.06/GB in 2010. (Unfortunately, memory sticks at our favorite electronics stores are still a bit more expensive).

Changes, including lower cost of storage and lower cost of data recording devices undoubtedly, fuel the Big Data phenomenon and raise the question, "How do we (the PHM Community) extract meaning from that much information". Another question might be "What is the value of Big Data". One institutive value of more and more data is simply that statistical significance increases. This is certainly the case in data-driven prognostics. Yet, care is required. Consider the gold mine metaphor, where in the mine, only 20 percent of the gold is visible. The remaining 80 percent is in the dirt where it cannot be seen. Mining is required to realize the full value of the contents of the mine. Hence Big Data Analytics and data mining are required to achieve new insights that have never before been seen.

To fully characterize Big Data, consider Figure 2. The challenges of big data are variety, velocity, and volume. These three are often referred to as the three "V"'s of big data. Here we consider three additional V's, veracity, value, and visibility. Volume is the amount of data as measured in its computer disk or computer memory size. Velocity is the speed at which data is produced, and moved into the computing infrastructure. Veracity is a measure of accuracy or reliability of the data, in other words the validity of data. Variety is both the data structure such as binary files and database tables, and the sources such as vibration, temperature, and maintenance records. Value is the information and business guidance that can be extracted from the data. Last but not least, visibility is the ability to access and view data and its value, regardless of the location of the data within the computing infrastructure.

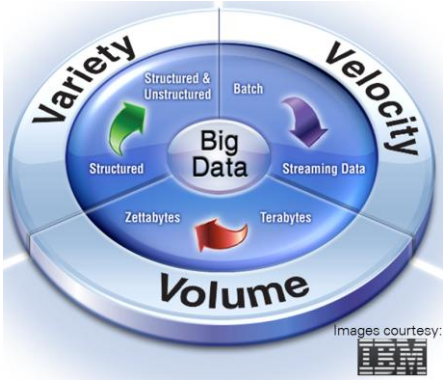


Figure 2. Traditional 3 "V"s of big data (source: IBM)

3. INDUSTRIAL INSTRUMENTATION, BIG DATA, PROGNOSTICS

The sources of Big Data in the Industrial Asset Monitoring arena are many, Figure 3. The most interesting is data derived, using transducers, from the physical world. In other words, this is analog data captured by instruments and data acquisition systems from a variety of vendors, in a variety of formats. Thus, the PHM community may call it "Big Analog Data" (BAD). BAD is derived from time waveform measurements from vibration, dynamic pressure, thermal images, ultrasonic scans, motor current signatures, and even radio frequency measurements used in the detection of partial discharge or electrical ground faults. Engineers, Scientists, and our plant Maintainers publish this kind of data (BAD) voluminously, in a variety of forms, and many times at high velocities. Along with management and storage of this large amount of data, are the challenges of validation or veracity, deriving value from the data, and giving visibility of data and derived value to the right people at the right time.

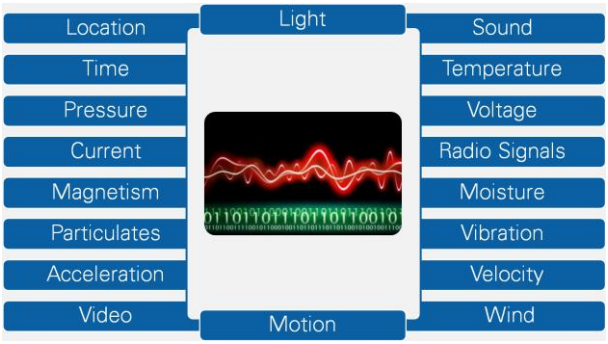


Figure 3. Industrial sources of analog data

As scientists and engineers work to address this "BAD" challenge, an approach is needed that encompasses sensors and actuators, distributed acquisition and analysis nodes (DAANs), and Information Technology (IT) infrastructure for big data analytics, mining and storage. Consider a three-tier solution, Figure 4. Here, it is possible to distribute the work of finding value in big analog data. Figure 4 depicts a three-tier architecture with sensors (and monitored assets) on the left. Measurement hardware or data acquisition systems are in the middle. These devices digitize analog sensory data from a single monitored asset and begin preliminary analysis. The right side of Figure 4 depicts the IT infrastructure employed to store, manage, and analyze sensory data from a fleet of assets.

Two additional terms are introduced here to describe veracity and extraction of value: "In-Motion" and "At-Rest" analytics. With In-Motion analytics, data is analyzed for value in the form of indicative information, in memory, and as close to the source of the data as possible. With At-Rest analytics, data is analyzed in its storage place often

incorporating similarities and differences with collaborative data sources. Both the DAANs and the IT computers perform in-motion analytics, extracting condition indicators. The IT infrastructure, as it assembles sensory and other data from multiple sources, also performs at-rest analytics utilizing data-driven prognostic algorithms to identify patterns and fault signatures.

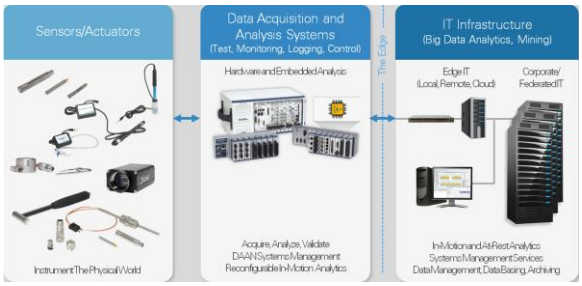


Figure 4. A three-tier solution to the "Big Analog Data" challenge.

Let's look closer at in-motion analytics close to the sensor. For example, adding a smart chip such as a Field Programmable Gate Array (FPGA) or a processor to an analog sensor allows the sensor to reduce the raw analog data to condition indicating features of the time waveform. However, it is also possible to add "smart" data recorders to the traditional analog sensors installed today. Both the smart sensor and the smart recorder are able to implement a decision based data recording technique, Figure 5. Here, analog sensory time waveform data is continuously analyzed for changes. Only when an indication of change within the asset is present in the sensory data (or on a time basis for periodicity) is the data recorded and forwarded upstream in the three-tier architecture. Further, the sensory data might be reduced using in-motion analytics to a set of condition indicators or features, leaving the raw time waveform stored locally or discarded. The filtering process of looking for changes and reducing data to condition indicators plays a big role in managing volume, velocity, veracity, and value.

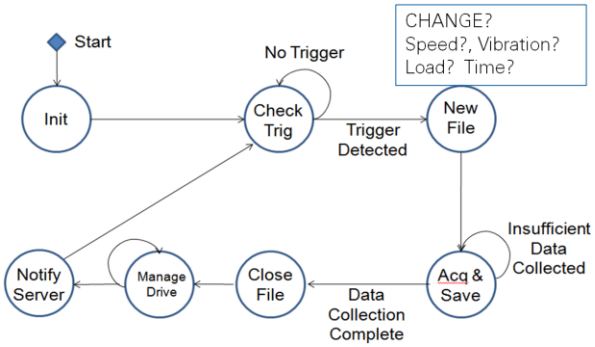


Figure 5. Decision based data recording state diagram

Whether, we have the ability to perform analysis in-motion at the sensor, at the DAAN or at-rest in the IT Infrastructure, we are fortunate to have a number of analytical tools at our disposal for finding value in the data. The scientific fields of condition monitoring and prognostics offer a number of analytical tools for reducing data to condition indicators and for finding trends in the analytical results, Table 1, Figure 6. Condition indicating analytics range from vibration level measurements, temperature trends, to envelope spectrum for roller bearing degradation and so on. With condition indicating analytics, we can discover increased impacting in roller element bearings, teeth cracking in gearboxes, rotor bar degradation in induction motors and generators, and so on. Condition indicators, coupled with trending and alarming, give the asset owner / operator a first alert that degradation is occurring within the asset.

Table 1. Condition indicating analytics

Graphic	Signal Characteristic	Analysis Methods	Machine Example
	Narrow frequency band lasting for a long time	Frequency Analysis Fourier Transform Power Spectrum	Unbalance in a single speed machine
	Narrow frequency band with harmonics lasting for a long time	Quefrency Cepstrum	Damaged bearing in a machine with roller element bearings
	Time varying frequency band	Time-frequency analysis Order analysis	Unbalance in a variable speed pump
	Wide frequency band signal lasting for a short time	Wavelet analysis AR Modeling	Low speed machine with compressor valve impacts
	Narrow frequency band signal lasting for a short time	Wavelet Analysis	Electrical motor driven machine with rub and knock noise.

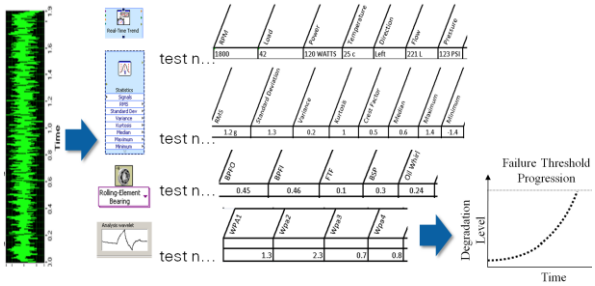


Figure 6. Reducing sensory data to condition indicators

Within the PHM community, the use of multiple condition indicators in concert, and an extensive history of actual condition indicators, data driven prognostics is made possible. Prognostic analytics include clustering, statistical pattern recognition, logistic regression, support vector machine, neural networks and so on. These are similar mathematics used in big data sciences, a growing profession and industry sector. Together, these two classes of analytics (condition indicators and prognostics) provide the foundation for finding value in big analog data. Long term, these tools are building the foundation for automating diagnostics, and prognostics. With the automation of diagnostics and prognostics, business decisions can be

enhanced with automatically generated advisories for maintenance, operations, and finance.

The condition indicators themselves do not necessarily yield a root cause for the degradation, nor does the condition indicator tell us when we can expect the asset to fail to perform its function. Prognostic analytics are employed to help deduce the why and when of asset degradation and failure, Figure 7.

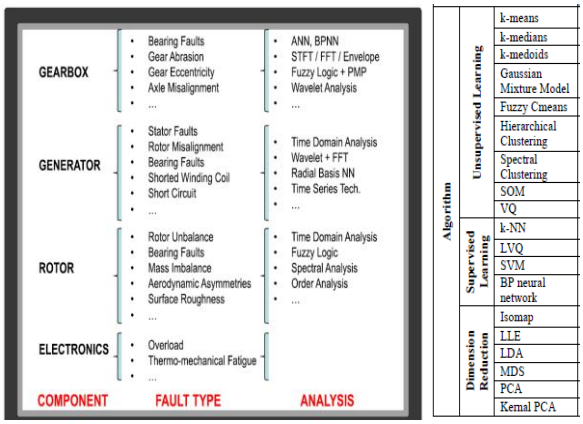


Figure 7. Prognostic analytics for finding patterns

Prognostic algorithms allow for the combination and collaboration of condition indicators within an asset (bearing, gear, shaft, oil particle, temperature, load, speed) as well as across similar assets. This combination of condition indicators forms a pattern of healthy asset operation, or a specific degradation pattern. In practice, a baseline of healthy condition indicators is obtained during commissioning of an asset, or after repair and maintenance of an asset. With an available healthy or normal operation pattern, analytical tools including statistical pattern recognition can be used to determine electrical, mechanical, or structural degradation levels of an asset, Figure 8. These tools compare real-time sensory data in-motion to patterns looking for deviations or anomalies.

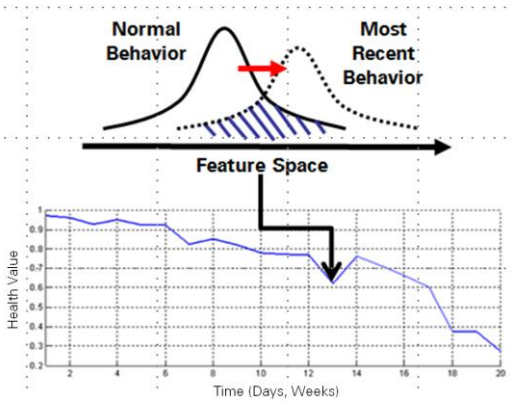


Figure 8. Asset degradation using statistical pattern analysis

The normal and fault patterns are further extended, by further segregating these patterns into operating conditions when speeds, loads, and environment are included. The combination of patterns at a plant or enterprise level, is made possible when similar assets are viewed together, enhancing the pattern formation. For example, machine learning algorithms are able to cluster combinations of condition indicators from similar assets, thereby creating patterns of normal or fault asset behavior. Prognostic algorithms then use these patterns, or fault signatures, to match current asset condition indicators to a specific fault signature (with in-motion analytics).

On another note, as condition indicators are narrowed in number to the best indicators of specific failure modes, a smaller set of sensors and analytics may be used to detect and predict specific failure modes. These reduced sensory measurements and analytics can then be performed on sensory data in-motion on the (embedded) DAAN, comparing a single vector of condition indicators to specific fault patterns.

As the normal operational pattern "drifts" towards a specific fault signature pattern, the rate of "drift" combined with human expert knowledge to form a basis for automatic advisory generation and prediction of the point in time when the asset fails to perform its function. This is particularly true at the information technology (IT) level, when future operating conditions are known based on planned equipment operations. Knowledge of a future operating condition allows focus on data-driven patterns from historical and specific expected operating conditions. Trends derived from historical specific operating conditions, improve confidence in the expected performance and health of specific equipment in planned operating conditions. At the plant or even enterprise level, the fusion of operational and equipment data builds a foundation for and confidence in the data-driven predictions.

To summarize, there are many physical phenomenon to measure within a fleet of assets. This creates the big data problem of the analog kind. By using in-motion and at-rest analytics, the six V's of big analog data are addressed. Analytics that calculate condition indicators, derive patterns of condition indicators, and compare real-time condition indicators to normal and faulty patterns are core to addressing the challenge of big analog data. This challenge of big analog data is deriving value and visibility while managing volume, velocity, veracity, and variety.

4. INFORMATION TECHNOLOGIES

In addition to sensory data, condition indicators, and asset operational patterns, we (the PHM community) often add other data which may be unstructured in nature. Work order reports, typed textual descriptions, and diagnostic technical

exams add to our big analog data, extending our view of the health of assets. To support big analog data storage and analytics as well as varied documentation, consideration and collaboration with our colleagues in Information Technology (IT) is a must.

Part of our challenge with big analog data and the varied documentation formats, is the data does not fit easily into standard relational databases. As a comparison, neither does the vast information available on the world-wide web. Out of Google's work to "index" the web, came an underlying file system, Apache Hadoop, which supports unstructured data or data that is stored in files rather than a relational database, Figure 9. These files can include binary and ASCII formats of condition indicators and time waveforms. Our unstructured data files also include asset technical exam documentation. There are many common formats used for big analog data including UFF58, Comtrade, and .mat. In the case study presented later, the file structure named Technical Data Management Streaming (TDMS) is used for storing time waveforms and condition indicators. The Apache Hadoop File System (HDFS) helps to manage these non relational database items. The HDFS is a massively scalable storage and batch data processing system. It provides an integrated storage and processing fabric that scales horizontally with commodity hardware and provides fault tolerance through software. Hadoop also includes concepts for distributing analytics to the data, to avoid bandwidth issues of moving the at-rest data (Bisciglia, 2009).

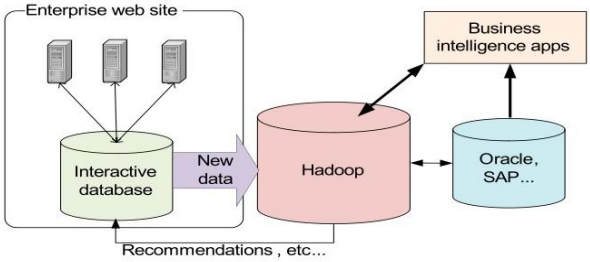


Figure 9. High level overview of Hadoop file system within IT architecture (source: Cloudera)

Several information technologies suppliers take the concept further by industrializing HDFS and improving the programming tools used to mine and analyze the data in a combination of Hadoop and relational stores. International Business Machines (IBM) for example, not only hardens the IT infrastructure with their "PureFlex" enterprise computing systems, IBM also adds InfoSphere Streams for in-motion analytics and InfoSphere BigInsights for at-rest analytics, Figure 10. These architectures and analytic tools promise an ability to quickly garner value of our variety, velocity and volume of Big Analog Data and unstructured documentation (Franklin, 2012).

The convergence of pervasive sensory data sources, new information technologies, growing information stores and a reduction in the overall cost and time needed for analysis has helped big data and specifically our industrial big analog data cross the chasm from innovation to early adoption. Big data is still an early-stage technology, but expect that over the next 18 months it will break double digits on project adoption basis. (Rogers, 2011).

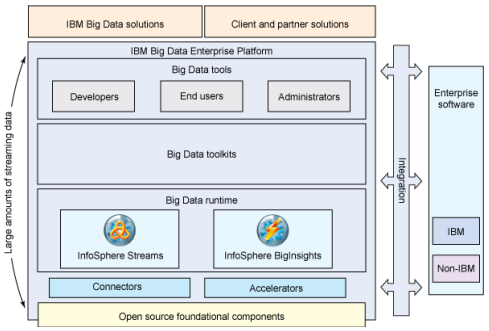


Figure 10. IBM's platform and vision for big data (source IBM DeveloperWorks)

So, if we can combine big analog data, in-motion and at-rest analytics of the condition indicating and prognostics kind, with expanded information technologies; perhaps it becomes possible to create smart monitoring and diagnostics, or even cloud based prognostics. The Center for Intelligent Maintenance systems projects a future where multiple end users will submit their asset data and condition indicators to a cloud resource (IMS, 2012) Here, analytical collaboration occurs to build and leverage fault signatures, degradation patterns, along with prognostic analytics to advise us on the current and future health of our assets, Figure 11.

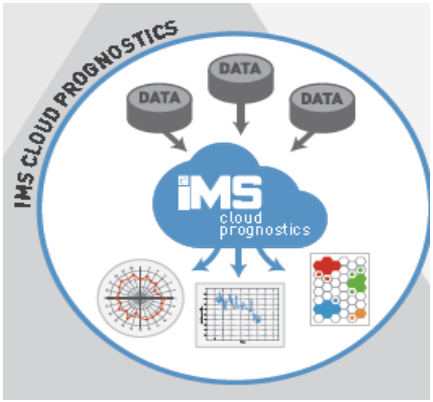


Figure 11. Center for Intelligent Maintenance Systems Cloud Prognostics Vision (source: IMS Center)

Given that Moore's law of big data is a true observation, then the doubling of data every two years demands that these information technologies will mature and become more pervasive. The field of prognostics will benefit from

the collaboration that comes with a wide net of assets, sensory data, and condition indicators derived from the sensory data. The combination of prognostics and data science technologies with information systems technologies is already yielding solutions for the volume, velocity, veracity, variety, value, and visibility of the fleetwide monitoring big analog data challenge.

5. CASE STUDY

In power generation, the above mentioned technologies are coming together to solve fleetwide asset monitoring data and information challenges. The Electrical Power Research Institute (EPRI) continues to sponsor a fleet wide asset monitoring project within a special working group, the Fleetwide Monitoring Interest Group (FWMIG) (Hollingshaus, 2011). This program aims to articulate a condition based maintenance and prognostics solution for its power generation members. The applications framework leverages data available within power generation plants, a fault signature database, and traditional monitoring and analysis techniques for rotating machinery.

Duke Energy, an EPRI member, is already deploying hundreds of new low cost "smart" data acquisition and analysis nodes (DAAN) within several power generation plants (Cook, 2013). These DAANs use traditional piezoelectric dual mode accelerometers with temperature sensing elements to monitor for changes in balance of plant equipment that supports turbine generators, Figure 12.

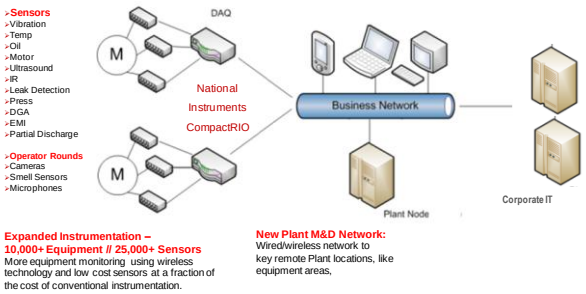


Figure 12. Duke Energy architecture for data acquisition and analysis nodes.

In the late 1990s, Duke Energy began its fleetwide monitoring program using commercial handheld instruments for vibration, thermography, ultrasonic, motor current, and oil analysis. Today, Duke Energy machinery health subject matter experts spend 80 percent of their time with these hand held instruments simply collecting sensory data.

Beginning in 2012, Duke Energy began to automate data collection with flexible DAANs, thereby reducing the labor costs and sparse periodicity associated with manual analog data collections. With the new DAANs in place, these same subject matter experts will be able to spend 80 percent of their time analyzing sensory data and planning maintenance actions. While the core initial motivation and return on investment at Duke Energy is employee utilization, the opportunity for prognostics, especially data driven, is tremendous as vibration, temperature, and oil analysis analog data now stream at regular intervals into the Duke Energy IT infrastructure, Figure 13.

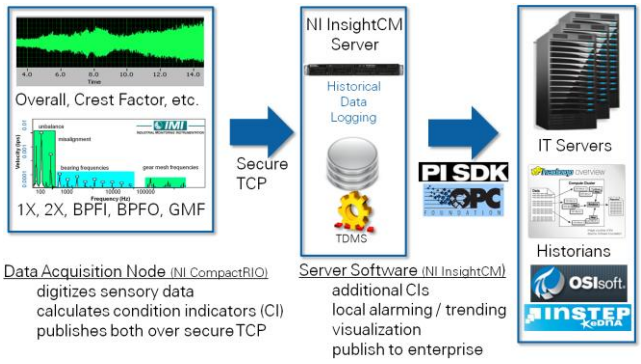


Figure 13. Big analog data sensory data flow

To accomplish the high level architectures, Duke Energy is working with EPRI and condition monitoring vendors to develop and implement a big analog data system for fleetwide asset monitoring that manages the six "V" challenges of big data. As shown earlier in Figure 5, and in Figure 13, the DAAN works to address volume, velocity, veracity, variety, and value. Using an event base local recording structure, Figure 5, sensory data is filtered to just data that is periodic or has a change. This filtering helps address volume. Using a store and forward communications scheme, data is transferred at the bandwidth allowed on the network. By storing and forwarding, the velocity of data is controlled by network administration tools. The DAAN also checks sensor value validity by using range checking and open/short cabling issues. This sensor value check helps address veracity. Lastly, the DAAN labels all data with sensory data type, measurement characteristics, and equipment hierarchy down to the component where the sensor is attached. The labeling tasks helps address the variety of the various analog measurements made by the DAAN.

To support the new volume, velocity, and variety of data coming from the newly deployed DAANs, Duke Energy has formed an IT task force to develop a big analog data strategy. The goal of the task force is to maximize value and visibility in particular with respect to equipment maintenance, availability and reliability. The current organization of data analytics orchestrated by Duke Energy IT, EPRI, and vendors is show in Figure 14. Value and Visibility at Duke Energy are determined at the monitoring and diagnostics center in Charlotte, NC. Here all condition indicators and operational process parameters are recorded in OSIsoft PI™'s historian for advanced pattern recognition

and anomaly detection by Instep Software's PRiSM™ predictive analytics tools.

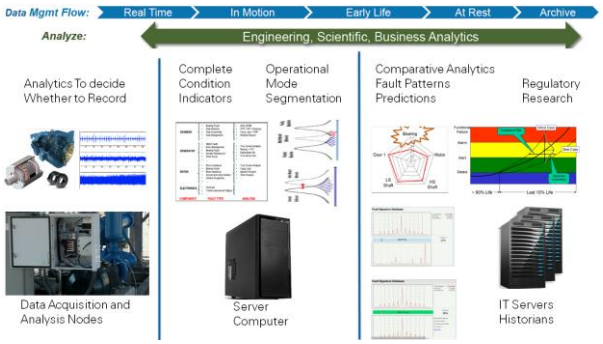


Figure 14. Analytics flow in big analog data applications

While the condition indicators are published to enterprise historians, the technical exam data including vibration time waveforms, stored in TDMS format, remains at the plant server level. This allows subject matter experts to access and analyze the analog sensory data using common graphics and analysis techniques associated with the particular technology. For example, vibration time waveforms are analyzed with frequency spectra, in the order domain, using harmonic, sideband cursors, and waterfall displays. The vibration analytical tools also provide trends and alarms at the local plant level for harmonics of rotational speed or order analysis, as well as trending of all condition indicators calculated at the DAAN or the plant server computer level.

However, time waveform data is big data, and the volume needs management at the plant level. Once condition indicators are extracted and published to the OSIsoft PI™ historian, some of the time waveform data can be discarded. An aging strategy is implemented that removes all time waveform data, after five days with the exception of those time waveforms most close to peak power demand times of day, 8:00 AM, Noon, and 4:00 PM. In addition, any time waveform that was recorded due to a measurement value alarm is preserved. Subject matter experts can also mark specific data files for preservation as the need arises.

As condition indicators are analyzed in the historian, user notes regarding equipment, maintenance records, best practices, and recommended actions are also assembled from various data sources and locations within the Duke Energy information technology infrastructure (Hesler, 2010). The challenge lies in assembling, storing, and retrieving information both from fleetwide asset monitoring and also operating parameters, maintenance activities, and equipment component health. To address the challenge, Duke Energy has deployed EPRI's PlantView® software platform for managing power plant assets and developing condition status reports on plant equipment, Figure 15.

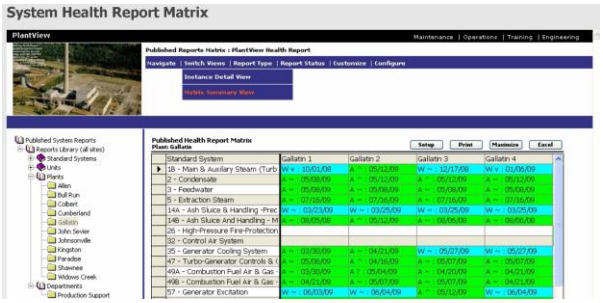


Figure 15. PlantView® health report matrix, image courtesy of Power Vision, Inc.

The PlantView software provides applications for entering storing and viewing information about plant operating parameters, maintenance activities, and equipment health. The status of equipment is kept in an integrated database. Visibility is provided thru a series of web services applications allowing users to access information from user customizable web portals. Duke Energy now has over 10,000 internal users benefiting from the PlantView web portals.

At Duke Energy, this is an obvious case where the opportunity for prognostics and IT come together to mine big analog data for the benefit of asset owners, asset operators, and the evolution of prognostics. Beginning with the DAAN, condition indicators extracted from monitored equipment, are supplemented with additional condition indicators at the plant server computer. This is the same computer that manages the DAANs. Subsequent to publishing the condition indicators to the enterprise historian, the advanced pattern recognition software begins comparison of current condition indicators to baselines for the specific operating condition. A web interface is provided for systems users and business owners to see both power output from generating units, as well as any equipment or process problems that may need addressing. The web interface, PlantView, brings the value and visibility of operations data to those responsible for making business decisions.

6. CONCLUSION

Big data, especially of the analog kind, can and does present challenges. Fortunately, information technology is evolving as quickly as the volume of data grows. Both in-motion and at-rest analytics are working to make sense of big analog data. The growing deployment of a wide range of sensors across a wide net of assets promises to accelerate the success and science of prognostic applications for monitoring fleets of assets.

REFERENCES

Bisciglia , C. (2009). 5 Common Questions About Apache Hadoop. *Cloudera Blog*, 14 May 2009

http://blog.cloudera.com/blog/2009/05/5-common-questions-about-hadoop/

Bradicich, T. & Orci, S. (2012). Moore's Law of Big Data *National Instruments Instrumentation News*. December 2012. Web. http://zone.ni.com/devzone/cda/pub/p/id/1649

Center for Intelligent Maintenance Systems (IMS), (2012). IMS Center brochure. *IMS Center website*. 5 Aug 2012. http://www.imscenter.net/Resources/brochure_2012_red_final.pdf

Cook, B. (2013). Deploying smart maintenance and diagnostics for electrical power generation. *NIWeek 2013*. 7 Aug 2012. https://decibel.ni.com/content/docs/DOC-30892

Franklin, C. (2012). Big Data as part of an enterprise data strategy. *Tamgroup Blog*. 19 March, 2012 http://www.tamgroup.com/blog/bid/118927/Big-Data-as-part-of-an-enterprise-data-strategy

Gantz, J., & Reinsel, D. (2011). Extracting value from chaos. *EMC Corporation website*. June 2011. Web.

Hadhazy, A. (2012). Zettabytes now needed to describe global data overhead. *Live Science*. 4 May 2010 Web.

Hessler, S. & Noce, G. "New web applications in EPRI's PLantView software offer Progress Energy enhanced capabilities for using plant data", publication 1021286, Electrical Power Research Institute, Palo Alto, California, USA

Hollingshaus, B. (2011). Program 69: Maintenance management and technology. *Electrical Power Research Institute Descriptions of Past Research.* Catalog number 1022681, May. 2011 www.epri.com

Johnson, P. & Douglas F. (2011). The impact of rapidly changing computing technologies on prognostic asset management applications. *MFPT Newsletter*. November 2011. Web. http://www.mfpt.org/Newsletters/1111/Johnson.htm

Losito, R. (2011). World's largest particle accelerator" *National Instruments Case Study*. 2011. Web. http://sine.ni.com/cs/app/doc/p/id/cs-10795

Rogers, S. (2011). Big data is scaling BI and analytics. *Information Management*. 1 Sep 2011.

BIOGRAPHIES

Preston Johnson is the Principal Sales Engineer for Condition Monitoring Systems at National Instruments (NI) in Austin, Texas. He has worked for National Instruments for over 27 years in roles of Field Sales, Sales Management, Automation Business Development, Sound and Vibration Segment Manager, Platform Manager for Condition Monitoring Systems and Global Program Manager for Asset Monitoring Systems. In his current role as Principal Sales Engineer, Preston works with NI OEM and End User customers to deploy fleetwide asset monitoring systems that lower operation costs, improve machinery reliability, and ultimately increase revenue. His interests lie in embedded signal processing, data acquisition systems and architectures, and prognostics. He earned his BSEE in Electrical Engineering and Computer Science from Vanderbilt University in 1985 and his MBA in Information Systems from the University of Texas in 1987. Preston is experienced in project management and holds a Category III vibration analyst certificate.

Fault Diagnosis and Prognosis Based on Lebesgue Sampling

Bin Zhang and Xiaofeng Wang

Department of Electrical Engineering, University of South Carolina, Columbia SC, USA
zhangbin@cec.sc.edu
wangxi@cec.sc.edu

ABSTRACT

Traditional fault diagnosis and prognosis (FDP) approaches are based on periodic sampling, *i.e.* samples are taken and algorithms are executed both in a periodic manner. As the volume of sensor data and complexity of algorithms keep increasing, the bottleneck of FDP is mainly the limited computational resources, which is especially true for distributed applications where FDP functions are deployed on microcontrollers and embedded systems with limited computation resources. This paper introduces the concept of Lebesgue sampling in FDP and proposes a Lebesgue sampling based fault diagnosis and prognosis (LS-FDP) framework. In the proposed LS-FDP, a novel diagnostic philosophy of "execution only when necessary" is developed in computation cost reduction and uncertainty management. For prognosis, different from traditional approaches in which the prognostic horizon is on the time axis, the proposed approach defines prognostic horizon on the state axis. With a reduced prognostic horizon, the LS-FDP naturally benefits the uncertainty management. The goal is to create the fundamental knowledge for LS-FDP solutions that are cost-efficient, capable for the deployment on systems with limited computation sources, and supportive to the trend of distributed FDP schemes in complex systems. The design and implementation of LS-FDP based on particle filtering algorithms are presented with experimental results to verify the effectiveness of the proposed approaches.

1. INTRODUCTION

Integrated System Health Management is a critical capability required for many safety critical systems such as unmanned air/ground/sea vehicles, aircraft, power generation, nuclear power plants, and various industrial systems (Tang, Zhang, DeCastro, & Hettler, 2011; Tang, Hettler, Zhang, & DeCastro, 2011; DeCastro, Tang, & Zhang, 2011; Zhang, Tang, DeCastro, & Goebel, 2011; Balanban & Slonso, 2013). The fundamental enabling technologies of integrated system health management include sensing, data acquisition, fault diagnosis and prognosis (FDP), and decision-making, etc. Diagnosis and prognosis, as fundamental enabling techniques, are not new concepts (Tumer & Bajwa, 2004; Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006; Zhang, Khawaja, Patrick, & Vachtsevanos, 2008; Schwabacher & Goebel, 2007). Diagnosis aims to monitor the health state of the component or the system such that the current health state can be obtained in real-time. The challenge in diagnosis is to detect potential faults as early and accurate as possible during the operation of a monitored system. Usually a fault cannot be measured directly. In Bayes theory, the fault state can be obtained by applying Bayesian estimation with a fault diagnostic model and a real-time measurement (Boskoski & Urevc, 2011; Zhang, Khawaja, Patrick, & Vachtsevanos, 2010; Zhang, Sconyers, et al., 2009; Zhang, Khawaja, et al., 2009; Li, Kurfess, & Liang, 2000; Goebel, Eklund, Hu, Avasarala, & Celaya, 2006; Goebel, Saha, & Saxena, 2008). In the context of fault diagnosis, the real-time measurements are often features or fault condition indicators extracted from raw measurements, such as vibration, current, voltage.

Prognosis refers to the generation of long-term predictions that describe the evolution of a fault and the estimation of the remaining useful life (RUL) of a failing component or subsystem. In reliability study, there are many diagnostic and prognostic approaches, such as Weibull-based risk distributions (Kaminskiy, 2005), the graphical reliability degradation modeling approach (Huang & Dietrich, 2005), and the degradation path curve approach (Lawless, 2003; Finkelstein, 2004; Yang, 2005), to name a few. For online prognosis, filter-based approaches are more promising, such as Kalman filter (Celaya, Saxena, & Goebel, 2012), extended Kalman filter (Saha, Goebel, Poll, & Christophersen, 2009), unscented Kalman filter (Anger, Schrader, & Klingauf, 2012), and particle filter (Zhang et al., 2010). Compared with many successful cases of diagnosis (Isermann, 2005; Zhong, Fang, & Ye, 2007; Hess & Wells, 2003; Zhang et al., 2010; Zhang, Sconyers, et al., 2009; Zhang, Khawaja, et al., 2009; Zhang et al., 2008; Oppenheimer & Loparo, 2002; Agogino, Bonis-

Bin Zhang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

sone, Goebel, & Vachtsevanos, 2001; Jardine, Lin, & Banjevic, 2006), prognosis is more challenging (Schwabacher & Goebel, 2007; Vachtsevanos et al., 2006; Edwards, Orchard, Tang, Goebel, & Vachtsevanos, 2010; Usynin & Hines, 2007; Celaya et al., 2012). Major contributors to this difficulty include nonlinear nature of fault growth, absence of measurement, hybrid nature of fault modes, and various uncertainties.

A comparison of several prognostic approaches can be found in (Goebel et al., 2008). To evaluate the performance of FDP, different performance indexes were also developed (Saxena, Celaya, Saha, Saha, & Goebel, 2009; Orchard & Vachtsevanos, 2009). For diagnosis, the matrices are often related to false alarm rate, probability of detection, etc. For prognosis, most matrices are evaluated in terms of accuracy and precision of RUL estimation. These metrics are often offline evaluated when failure has been physically reached and is compared with the RUL estimation from prognosis.

Traditional ways to design FDP algorithms adopt periodic sampling (also called "Riemann sampling (RS)") where samples are taken in a periodic manner and the diagnostic and prognostic algorithms are executed at the same rate. A nice feature of FDP with this fixed time interval sampling is the easiness in analysis and design. However, it may be undesirable in many situations, from the computation-efficiency point of view. On the one hand, since the sampling period is determined according to the worst-case operating scenario, the FDP algorithm might be executed even if there is little new information actually present in the measurements. In other words, the algorithm may take greater utilization than it actually needs. This will result in significant overprovisioning of the real-time system hardware. On the other hand, when the fault grows very fast, it is expected to assign more resources to the FDP algorithm so that it can takes more frequent actions to provide accurate fault information, which obviously cannot be met by periodic sampling. For prognosis, RS-based FDP usually has a large prediction horizon, from the time that a fault is detected at very early stage to a future time instant that the fault grows to the failure threshold. This long-term prediction not only requires a lot computation resources, but also causes accumulation of uncertainties. The LS-FDP considers the prediction horizon in the fault dimension axis and described by the number fo fault states. This provides a straightforward means to conduct prognosis that requires little computation resources.

As the applications of FDP has increased rapidly, the heavy demand on computational resources makes existing FDP algorithms very hard to be deployed on embedded systems that are widely used but have very limited computation capabilities. This becomes the bottleneck that prevents the distribution of FDP algorithms in complex systems. To break this bottleneck, cost-efficient FDP solutions must be developed. With this vision, we propose the Lebesgue sampling-based FDP (LS-FDP) method, which is a cost efficient FDP approach where computation can be executed on an "as-needed" basis and is promising in reducing the computational cost compared with the traditional Riemann sampling-based FDP (RS-FDP) algorithms. In this new approach of FDP, the novelty comes from the concept of "Lebesgue sampling (LS)" (or "event-based sampling"). Contrast to conventional periodic sampling-based approaches, the computation in LS-FDP will be triggered only when an event takes place, and the prognosis will be executed based on the LS-based model whose states are predefined according to the quantization level. With the feature of "execution only when necessary" in LS, the computation efforts in LS-FDP can be significantly reduced by eliminating unnecessary computation when fault growth is slow.

The paper is organized as follows: Section 2 provides an overview of the proposed LS-FDP framework. Section 3 develops a particle filtering based LS-FDP approach, which is followed by experimental results on an epicycle planetary gear box presented in Section 4. Section 5 gives the concluding remarks with some future research topics.

2. THE PROPOSED LS-FDP FRAMEWORK

This section will establish the complete LS-FDP framework with an overview of the proposed solutions. The unique innovative feature of the proposed LS-FDP is that the diagnostic and prognostic algorithm is no longer carried out in a fixed time interval. Instead, the diagnosis is carried out only when new measurements justify that the fault conditions have changes to warrant the execution. The LS-FDP framework is illustrated in Figure 1, which integrates external inputs, Lebesgue samples of feature and fault dimension, models for diagnosis and prognosis, and diagnostic and prognostic algorithms.

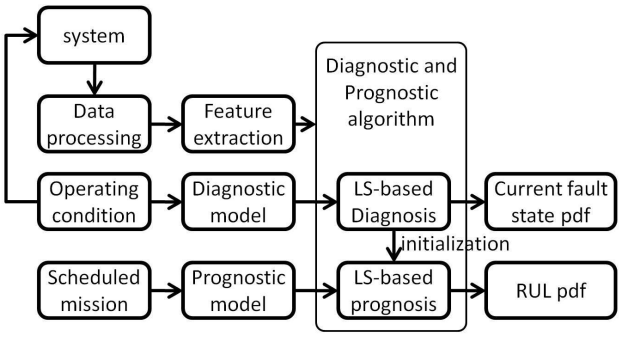


Figure 1. The implementation framework of LS-FDP

In this paper, our focus is the introduction of Lebesgue sampling into diagnosis and prognosis. Therefore, we will not discuss data collection, preprocessing, and feature extraction. After a feature has been successfully extracted from data to

indicate the growth of a fault, the performance and efficiency of FDP relies greatly on the dynamic model that describes the fault behavior, and the diagnostic and prognostic algorithms, which will be elaborated in the following sections.

2.1. Fault Mechanism Modeling

Assume that the actual fault growth dynamics can be described by the following continuous-time differential equation:

$$\dot{a} = F(a, u) \tag{1}$$

where a is the fault dimension, u is system input including items (such as external environmental factors and operating modes) that have impacts on fault growth, and $F(\cdot)$ is a nonlinear function that describes the fault growth under the current fault dimension with input u. The feature or condition indicator, denoted by y, is extracted from raw measurements and serves as the real-time measurement for FDP algorithm. Note that the mapping between y and a can be described by a nonlinear function $y = h(a)$. In most cases, a is not measurable and $y = a$ is employed such that we can use y to indicate fault a directly. To simplify the description, we take $y = a$ in the following discussion.

To use this model in LS-FDP, we quantify the fault measurements. Lebesgue sampling basically takes samples when the difference between the current state and the last sampled state exceeds the pre-defined Lebesgue state length. Then the LS-based model of the fault dynamics in discrete-time can be described as follows:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D, \dot{\hat{a}}(t_k)) \tag{2}$$

where $\hat{a}(t_k)$ is the Lebesgue state, t_k is the kth sampling instant, D is the Lebesgue length, and $f_t(\cdot)$ is a nonlinear function.

In traditional prognostic algorithm, there are two steps of prognosis. The first step is the generation of a long-term predication for the fault state pdf estimation. This is obtained by recursive execution of the fault growth model. The second step is the estimation of RUL, which is essentially related to the probability of failure at future time instants. The RUL pdf is obtained by defining a failure threshold established from historical data or empirical knowledge and comparing this threshold with the long-term prediction of fault state at all the future time instants. Compared to diagnosis, prognosis requires much more computational resources mainly because of long-term predication, especially when the prediction horizon is large, which is not a rare case in FDP applications. To reduce computation time and resources, a new model is developed in the LS-FDP as follows:

$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) \tag{3}$$

Note that $\dot{\hat{a}}(t_k) = f(\hat{a}(t_k), u(t_k))$ and $g_t(D, \dot{\hat{a}}(t_k))$ is a nonlinear function. Rather than conducting a long-term prediction on the time axis, this model calculates the RUL on each Lebesgue state directly so that the prediction horizon is the number of Lebesgue states on the fault dimension axis. Since the number of Lebesgue states on the fault dimension axis is small, the prediction horizon for LS-based prognosis is small and will significantly reduce the computation.

2.2. The Concept of Lebesgue Sampling

The concept of Lebesgue sampling can be illustrated through an example of a crack on a planetary gear carrier plate in a helicopter main power transmission system (Zhang et al., 2010). The seeded crack starts to grow from an initial value of 1.34 inches to 7.67 inches in 1000 cycles of operation and the ground truth crack dimension growth is shown in Figure 2. It is clear that the fault growth in the range $R_1 = [50, 650]$ cycle is slower than that in the range $R_2 = [650, 750]$ cycle. Using Riemann sampling-based FDP with fix time interval, as shown in Figure 2(a), the FDP algorithms are executed at each cycle no matter if it is necessary. Since the fix time interval is selected according to the worst-case scenario to guarantee tracking accuracy for fault growth in range R_2, there are many unnecessary calculations in range R_1.

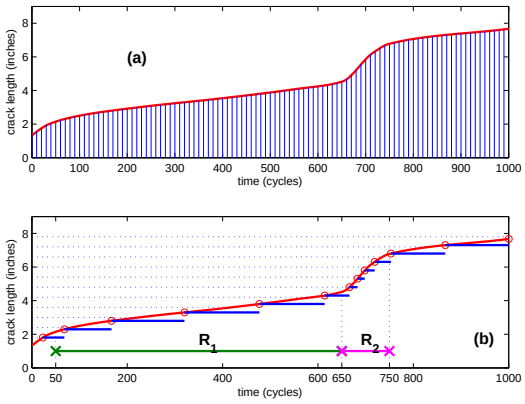


Figure 2. Illustration of LS. (a) RS with fixed time interval; (b) LS with fixed Lebesgue state length

Ideally, we expect to reduce the number of FDP execution in the range R_1 where the fault growth is slow so that more resources can be assigned to other tasks. In the range of R_2 where the fault growth becomes fast, we increase the number of FDP execution by assigning more resources to FDP tasks. This setting is desirable in FPGA-based embedded systems where resources are dynamically reconfigurable and are assigned to different tasks in realtime. With this configuration, a balance between computation and performance can be achieved. This strategy however involves time-varying sampling periods that is not an easy task within the Riemann sampling framework. With Lebesgue sampling, the realization of

this strategy becomes natural. By defining Lebesgue states on the vertical axis of fault dimension (crack length in this figure), fewer transitions between states are made when the fault growth is slow while more transitions are made when the fault growth is fast. For the example shown in Figure 2.(b), only 5 Lebesgue states are visited during the 550 cycles in R_1 and 4 states during the 100 cycles in R_2, which means that the FDP only needs to be executed 5 times during R_1 and 4 times during R_2. With this consideration, during R_1, more computation resources can be assigned to other tasks while only a little resources are needed for FDP. During R_2, more resources are assigned to FDP tasks so that the fault dimension can be tracked accurately.

2.3. Lebesgue Sampling-Based Diagnosis

In the LS-FDP framework, the range of the state $a(t)$ is partitioned into Lebesgue states $\{F_1, F_2, \cdots, F_f\}$, with which the diagnostic model is discretized. The diagnostic algorithm is executed when an event happens, *i.e.* the state $a(t)$ changes from one Lebesgue state to another one (McCann & Le, 2008; Astrom & Bernhardsson, 1999). The time instant when an event is generated is called the "event stamp". The sequence of the event stamps is denoted as $t_1, t_2, t_3, \cdots$, which formulates a time series that can be used as the input of run-time diagnostic algorithms such as a Kalman filter-based or particle filter-based algorithm (Morales-Menendez, de Freitas, Monterrey, Freitas, & Poole, 2002; de Freitas, 2002; Zhang et al., 2010; Zhang, Sconyers, et al., 2009; Orchard, Hevia-Koch, Zhang, & Tang, 2013). The output of diagnostic algorithm is the current fault state distribution at these event stamps and the probability of fault detection. The implementation procedure of the Lebesgue sampling-based diagnosis can be illustrated in the flow charts shown in Figure 3.

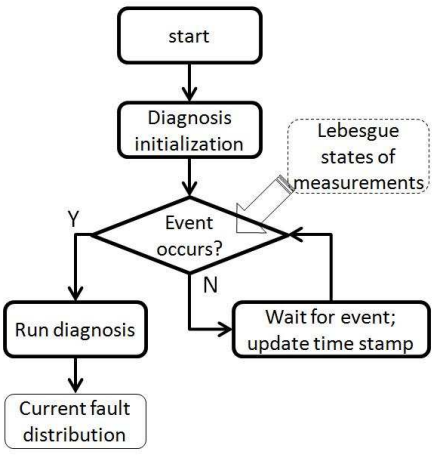


Figure 3. Flow chart of Lebesgue sampling-based diagnosis

2.4. Lebesgue Sampling-Based Prognosis

When a fault is detected at t_d, a time distribution is initialized as the initial condition for prognosis. By Riemann sampling-based prognosis, the prediction is conducted from the current time instant $t_{current}$ to future time instants till t_{fail} when the fault state reaches a failure threshold F_f. The prognostic horizon $[t_{current}, t_{fail}]$ is usually large, especially at the early stage of the fault or when the fault growth is slow. The prediction calculates the fault state at each fixed time interval, which is demanding on the computational resources. Moreover, prognostic uncertainty will grow rapidly with large prediction horizon.

With LS, a new prognostic philosophy is proposed. Suppose that the fault is detected at Lebesgue state F_d, then we consider the discretized prognostic model with Lebesgue states $\{F_d, F_{d+1}, \cdots, F_f\}$. The prognostic algorithm is implemented, together with the LS-based prognostic model, to calculate the distributions of operation time when the fault state reaches different Lebesgue states $\{F_d, F_{d+1}, \cdots, F_f\}$. Meanwhile, it will provide a RUL estimation on Lebesgue state F_f. Note that the prognostic horizon can be controlled by adjusting Lebesgue state length. Increasing the Lebesgue state length will decrease the number of events, which will reduce the required computational resources. The implementation procedure of the Lebesgue sampling-based prognosis can be illustrated in the flow charts shown in Figure 4.

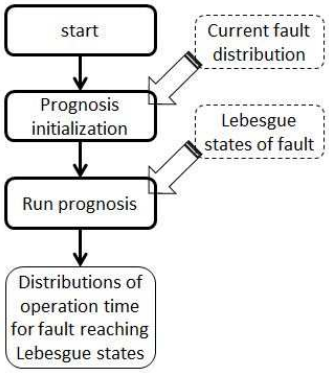


Figure 4. Flow chart of Lebesgue sampling-based prognosis

3. METHODOLOGY DEVELOPMENT

3.1. Particle Filter for LS-Based Diagnosis

The fault diagnosis is basically a state estimation problem, which can be handled in a Bayesian framework. Mathematically, assume the unobserved fault process X to be a Markov process characterized by initial distribution $p(x_0)$ and the transition probability $p(x_k|x_{k-1})$ defined by $x_k = f_k(x_{k-1}, \omega_k)$ with ω_k being the process noise. The subscript k represents the kth event stamp caused by the transition of Lebesgue

states. The observations Y are assumed to be conditionally independent given X. The distribution of $(Y_k|X_k)$ is defined by $y_k = h_k(x_k, v_k)$ with v_k being observation noise. Let $x_{0:k} = \{x_0, \cdots, x_k\}$ and $y_{1:k} = \{y_1, \cdots, y_k\}$ denote the state and the observation up to the kth event. It is of interest to estimate the *posterior distribution* $p(x_{0:k}|y_{1:k})$. The task can be achieved by two sequential steps, prediction and filtering.

In most nonlinear cases, however, analytical solutions do not exist. Alternatively, sequential Monte Carlo (SMC) methods, such as particle filter (Zhang et al., 2010), provide approximate solution to state estimation that is used for fault diagnosis.

Assume that a set of N particles $(w_{k-1}^{(i)}, x_{0:k-1}^{(i)})$ is available such that they can be used to approximate a desired distribution $\pi_{k-1}(x_{0:k-1})$, where the superscript $i = 1, 2, \cdots, N$ denotes N particles located at $x_{0:k-1}^{(i)}$ and $w_{k-1}^{(i)}$ is the weight of the ith particle at the $(k-1)$th event. The objective is to efficiently obtain a new set of N particles $(w_k^{(i)}, \bar{x}_{0:k}^{(i)})$ that can approximate the distribution $\pi_k(x_{0:k})$, where $\bar{x}_{0:k}^{(i)}$ denotes location of N new particles. In the context of SMC methodology, a Monte Carlo approximation can be obtained as:

$$\pi_k(x_{0:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_{0:k} - \bar{x}_{0:k}^{(i)}\right). \tag{4}$$

with $\sum_{i=1}^{N} w_k^{(i)} = 1$, where δ denotes the Dirac-delta function. The weight can be updated in a recursive formula as:

$$\begin{aligned} w\left(\bar{x}_{0:k}^{(i)}\right) &= w_{k-1}^{(i)} h_k\left(y_{1:k}|\bar{x}_{0:k}^{(i)}\right) \quad \text{and} \\ w_k^{(i)} &= \frac{w\left(\bar{x}_{0:k}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\bar{x}_{0:k}^{(i)}\right)}. \end{aligned} \tag{5}$$

To implement the above mentioned particle filtering based fault diagnosis with LS, an LS-based diagnostic model is given by:

$$\begin{cases} \begin{bmatrix} x_{d,1}(t_{k+1}) \\ x_{d,2}(t_{k+1}) \end{bmatrix} = f_b\left(\begin{bmatrix} x_{d,1}(t_k) \\ x_{d,2}(t_k) \end{bmatrix} + n(t_k)\right) \\ \hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t\left(D, \dot{\hat{a}}(t_k)\right) \cdot x_{d,2}(t_k) + \omega_a(t_k) \\ y(t_k) = \hat{a}(t_k) + v(t_k) \end{cases} \tag{6}$$

with nonlinear mapping $f_b(x)$ is given by

$$f_b(x) = \begin{cases} [1 \quad 0]^T, & \text{if } \|x - [1 \quad 0]^T\| \leq \|x - [0 \quad 1]^T\| \\ [0 \quad 1]^T, & \text{otherwise.} \end{cases}$$

and the initial condition is given by:

$$\begin{bmatrix} x_{d,1}(0) \\ x_{d,2}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where $x_{d,1}$ and $x_{d,2}$ are a collection of Boolean states that indicate *normal* and *faulty* conditions, respectively, $\hat{a}$ is the Lebesgue state that represents the fault dimension, ω_a and v are process and observation noises, respectively, n is independent and identically distributed uniform white noise, and u is the external input. In this equation, t_k is the event stamp indicating that there is a state transition event. As assumed earlier, the feature value $y(t_k)$ indicates the fault value $\hat{a}(t_k)$ directly, in order to simplify the description.

During the process of LS-based diagnosis, the diagnostic algorithm is executed only when the new measurement y shows that significant information is included. For this purpose, the range of feature (also fault in this case) is divided into a series of Lebesgue states. If two successive measurements cause a transition of Lebesgue state, the diagnostic algorithm will be executed. Otherwise, it won't be executed.

3.2. Particle Filter for LS-Based Prognosis

Prognosis estimates the RUL. In traditional RS-based prognosis, the prediction is carried out with fix time interval from the current time instant $t_{current}$ to the time instant t_{fail} that fault state reaches failure threshold F_f. The particles are estimated at each future time instant to approximate a fault state distribution at that time instant (the first prognosis level). Then, the fault distributions at all the future time instants are compared with the failure threshold F_f by applying the law of total probability to calculate the RUL distribution (the second prognosis level).

This RS-based prognostic approach often involves a large prognostic horizon, especially at the early stage of a fault and when the fault growth is slow. This large prognostic horizon causes two major issues. First, it is computationally expensive and not suitable for applications with limited computational resource. Second, the uncertainty in prognosis is inherent and will accumulate as the prediction horizon increases. When the uncertainty becomes too large, the estimation of the RUL becomes unreliable that cannot be used in decision-making.

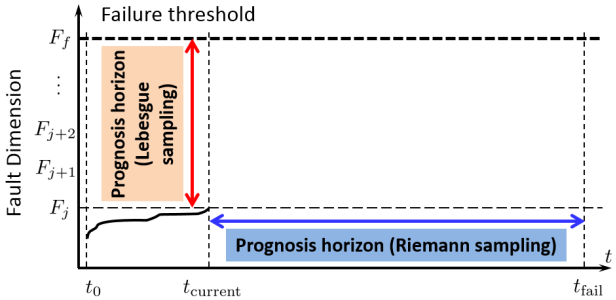


Figure 5. Comparison of prognostic horizon

With LS, the prediction horizon reduces to the number of Lebesgue states from the current Lebesgue state F_j to failure threshold F_f. With this idea, each run of the prognostic

algorithm guarantees that the fault has changed and an event has been generated. As a result, a large amount of unnecessary computation can be avoided, which is impossible with RS. It will not only reduce the requirements on computational resources, but also provide an intuitive way to manage uncertainties in prognosis. The comparison of prognostic horizon with RS and LS is illustrated in Figure 5.

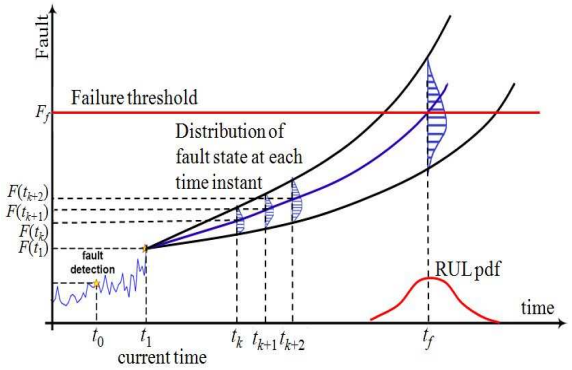
In the context of LS, the prognostic model is given by:

$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) + \omega_t(t_k) \tag{7}$$

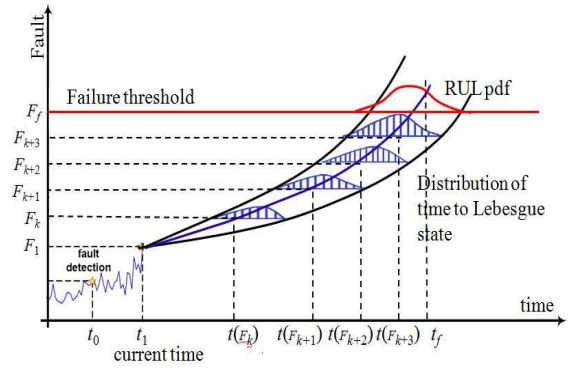
where D is Lebesgue state length and $\omega_t(t_k)$ is a model noise.

With this model, the particles are defined on the time axis instead of the fault dimension axis in RS-based prognosis. To initialize the prognosis, a new set of N particles is defined as $(w_L^{(i)}, t_L^{(i)})$, in which subscript L denotes the Lebesgue state, $w_L^{(i)}$ denotes the particle weight, and $t_L^{(i)}$ denotes particle on the time axis. The initial particles can be equally weighted with $w_L^{(i)} = \frac{1}{N}, \forall i$ or from diagnosis.

Note that the prognosis is carried out with a model given by equation (7). The outcome is the distributions of the operating time for the fault state to reach each Lebesgue state. Therefore, in this LS-based prognosis, the RUL pdf is calculated directly at the Lebesgue state $L = F_f$.



(a) RS-based prognosis



(b) LS-based prognosis

Figure 6. Comparison of RS-based prognosis and LS-based prognosis

The difference between RS-based and LS-based prognosis is illustrated in Figure 6. We assume that a fault is initialized at an unknown time instant t_0. The fault is detected at t_1 and prognosis is activated from this time instant. For RS-based prognosis in Figure 6(a), the prediction horizon is $[t_1, t_f]$, where t_f is the time stamp when the prediction of all particles pass the failure threshold. With a sampling period of T, the prognostic algorithm needs to recursively prediction all particles $(t_f - t_1)/T$ steps and this is the most time-consuming part of prognosis which limits many applications. In other words, the prediction steps are $[t_1, \cdots, t_k, t_{k+1}, t_{k+2}, \cdots]$ on the horizontal time axis. The expectations of the distributions of the operating time to reach these Lebesgue states are $[t_1, \cdots, t(F_k), t(F_{k+1}), \cdots, t_f]$, of which the time intervals could be uneven.

In the Lebesgue sampling-based prognosis, the prediction horizon is $[F_1,\ F_f]$ where F_f is the fault dimension that indicates the failure of the system. With a uniform Lebesgue length of D, there will be $(F_f - F_1)/D$ predication steps, and can be denoted as $[F_1, \cdots, F_k, F_{k+1}, \cdots, F_f]$ on the vertical axis. The expectations of the distributions of the operating time for the fault reaching these Lebesgue states are $[t_1, \cdots, t(F_k), t(F_{k+1}), \cdots, t_f]$, of which the time intervals are uneven. In summary, the fundamental difference is that RS-based prognosis calculates fault state distribution at given time instants, while LS-based prognosis calculates time distribution at predefined Lebesgue states.

4. EXPERIMENTAL RESULTS

In this section, the proposed LS-FDP scheme with a particle filtering algorithm will be verified in a case study of an epicyclic gear system in which a crack in the planetary carrier plate is developed.

4.1. Planetary Gear Box

The main transmission of Blackhawk and Seahawk helicopters employs a five-planet epicyclic gear system, which is a critical component directly related to the availability and safety of the vehicle. The fault is a crack in the planetary carrier plate, as shown in Figure 7.

A timely detection of crack and prediction of failure will not only help the decision-making on mission planning and system reconfiguration, but also improve the reliability and safety of the vehicle. In the experiments, a fault of seeded crack grows with the evolving operation of the gearbox. The gearbox operates over a large number of Ground-Air-Ground (GAG) cycles at different torque levels. An accelerometer is mounted at a fixed position to collect vibration data as crack length grows. In our previous research, the vibration signal processing and feature extraction have been discussed and applied in Riemann sampling based diagnosis and prognosis (Chen, Zhang, Vachtsevanos, & Orchard, 2011; Chen, Zhang,

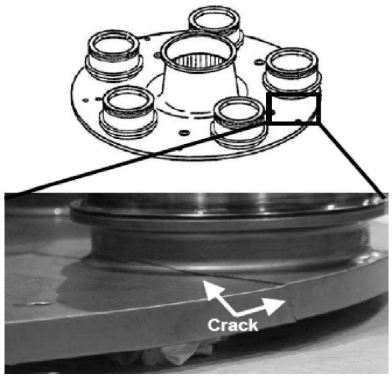


Figure 7. Crack of planetary gear carrier plate.

& Vachtsevanos, 2012; Zhang, Khawaja, et al., 2009; Zhang, Sconyers, et al., 2009; Zhang et al., 2008). In this section, we will use the crack growth data for verification of LS-FDP.

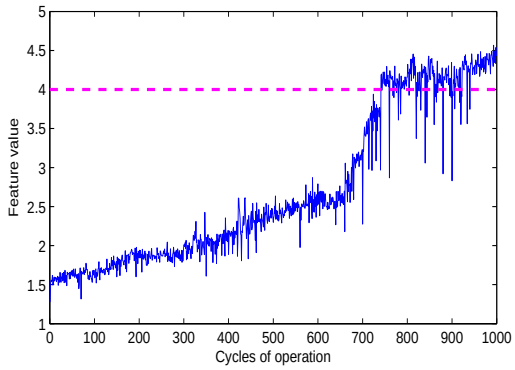


Figure 8. Feature vector for fault growth.

The feature vectors is shown in Figure 8. Since a fault is seeded in the experiment, the data from the first 50 cycles are used as baseline data, which has a mean value of 1.5741, and our objective is to detect the real-time fault growth from this baseline crack length. Note that this feature value will be used as a direct indicator of fault dimension as fault crack length as described in Section 2. For prognosis, the failure threshold is set as 4. The figure shows that the feature value reaches this threshold at around 750th cycle of operation.

4.1.1. RS-based Diagnosis and Prognosis

To implement diagnosis and prognosis, a fault growth model needs to be developed. For Riemann sampling based diagnosis and prognosis, the fault growth model is given by:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + p_1 \cdot a(t_k)^{p_2} + \omega(t_k) \tag{8}$$

where p_1 and p_2 are parameters and ω is a model noise.

A particle filtering with 500 particles are implemented and the results of fault diagnosis is shown in Figure 9. The fault is detected at the 183rd cycle, at which the expected value of fault state is 1.94 and the 95% confidence interval is $[1.79,\ 2.08]$.

In this figure, the top subfigure is the feature, given by blue curve, compared with the filtered feature, given by magenta curve. the bottom subfigure shows the comparison of baseline pdf (green one) compared with the real-time estimation pdf (red bars) at the cycle when the fault is detected. In this experimental, 5% false alarm rate is defined and the fault detection threshold is given by the blue vertical line. Note that in this RS-based diagnosis, the diagnostic algorithm needs to execute 183 time, *i.e.*, every time when a new feature becomes available.

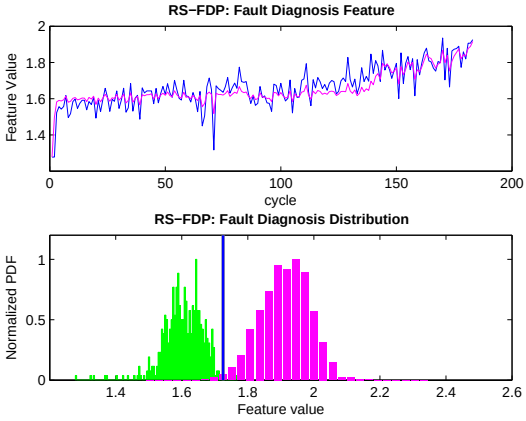


Figure 9. Experimental result of RS-based diagnosis.

As a fault is detected, prognostic algorithm is activated to conduct the long-term predication and estimation of RUL. The initial condition of prognostic algorithm is the fault state at the cycle when the fault is detected. The result of fault growth and RUL estimation is shown in Figure 10. To make the figure clear, only the fault state pdf at the 183rd cycle is plotted and the fault state pdf at other time instants are not shown in this figure. Instead, the expected value, upper and lower bound of 95% confidence interval of the pdf at each time instants are shown in this figure. Note that the prognosis needs to predict all particles from its current value at the cycle 183 to the failure threshold value. In this figure, the prediction horizon is about 700 cycles. To make the real-time implementation of prognosis possible, the number of particles is reduced to 20.

Then, the fault state pdf at each time instant is compared with the failure threshold to obtain the RUL pdf, as shown in the histogram on the horizontal axis. This process uses the law of total probabilities and can be mathematically described as:

$$p_{failure}(t) = \sum_{i=1}^{N} Pr\left(Failure|x_t^{(i)} > F_f\right) w_t^{(i)} \tag{9}$$

where superscript (i) is the index of particles, $p_{failure}(t)$ is the probability of failure at time t, $w_t^{(i)} = Pr(x = x^{(i)})$ is

the weight of particles at time t, and x_t is the predicted value of a particle at time t.

The RUL pdf is shown as the histogram in the figure. With this figure, the predicted expectation of the failure time is at the 588.6 cycle and the RUL life is 405.6 cycle. The 95% confidence bound of the RUL pdf is given as $[443\ \ 767]$. The uncertainty caused by the long prediction horizon is very large. In addition, from feature vector, we can see that the feature value reaches 4 at around 750th cycle. The distance from the predicted expected value to this ground truth value is 161.4 cycle.

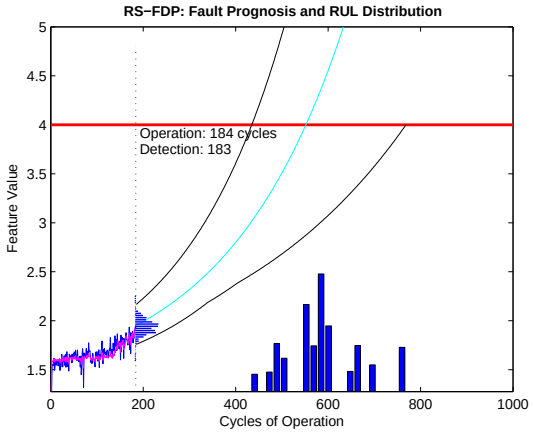


Figure 10. Experimental result of RS-based prognosis.

4.1.2. LS-based Diagnosis and Prognosis

For LS-based diagnosis, the feature value range [1.28 4.57] is partitioned into 20 states. The diagnostic algorithm is executed only when the collected feature value changes from one Lebesgue state to another, *i.e.* an event happens. The diagnostic model used in LS-based is given as:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + D \cdot \mathrm{sgn}(\dot{\hat{a}}(t_k)) + \omega_a(t_k) \qquad (10)$$

where $sgn(\cdot)$ is a sign function and ω_a is the model noise.

The diagnostic results are shown in Figure 11. In the particle filtering algorithm, 500 particles are used. The fault is detected at the 186th cycle. In the upper subfigure of Figure 11, the blue curve is the trajectory of feature values and the magenta curve is the filtered feature from particle filtering. Note that the flat segments mean no event and the diagnostic algorithm does not execute. The lower subfigure shows the fault distribution at the time of detection, where the green distribution is the baseline pdf while the magenta histogram is the real-time fault distribution from diagnosis. The blue vertical line is the threshold of fault detection with 5% false alarm rate. During these 186 cycles, there are 76 events, *i.e.*, the diagnostic algorithm only runs 76 times. The reduction of computational cost is 59.7%, which is a remarkable improvement. At the 186th cycle when the fault is detected, the expected value of fault state is 1.91 and the 95% confidence interval is $[1.69,\ 2.08]$.

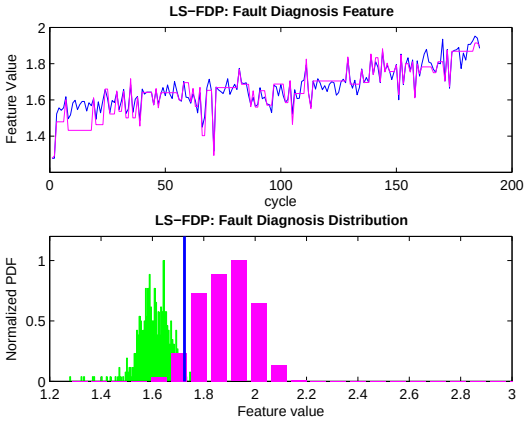


Figure 11. Experimental result of LS-based diagnosis.

Same as the RS-FDP, as the fault is detected, prognostic algorithm becomes activated. Since the prediction horizon is on the vertical axis, the initial condition of prognosis cannot use the estimation result from diagnosis directly. Therefore, we convert the fault state pdf at the time instant of fault detection to a time distribution for fault reaching the current Lebesgue state. This can be done by predicting those particle not yet reach the current Lebesgue state to this state. Then equation (9) is used to obtain the initial time distribution for prognosis. Note that for the prognosis shown in this figure, the prediction horizon is only 15 Lebesgue states, which is very small compare to that in RS-FDP, which is about 700 cycles. Therefore, the LS-FDP prognosis can afford the computation of 500 particles and we do not need to reduce the number of particles.

Since the prognosis is conducted on fault dimension axis, the diagnostic model cannot be used as we described in Section 2. The prognostic model used in LS-based prognosis is given as:

$$t_{k+1} = t_k + D \cdot exp(-\dot{\hat{a}}(t_k))) + \omega_t(t_k) \qquad (11)$$

The prognosis results are shown in Figure 12. To make the figure clear, only the time distribution pdf at a few selected Lebesgue state are plotted. Note that the time distribution pdf at the Lebesgue state defined by the failure threshold gives the RUL estimation pdf. In this figure, the predicted failure time is at the 689.4 cycle and the RUL life is 503.6 cycle. The 95% confidence bound of the RUL pdf is given as $[601\ \ 747.6]$. The uncertainty is much smaller than that of Riemann-sampling based prognosis. When the predicted RUL pdf expected value compared with the ground truth value of 750 cycle, the difference between them is 60.6 cycle.

The advantages of Lebesgue sampling in fault diagnosis and prognosis are obvious from the comparison of above exper-

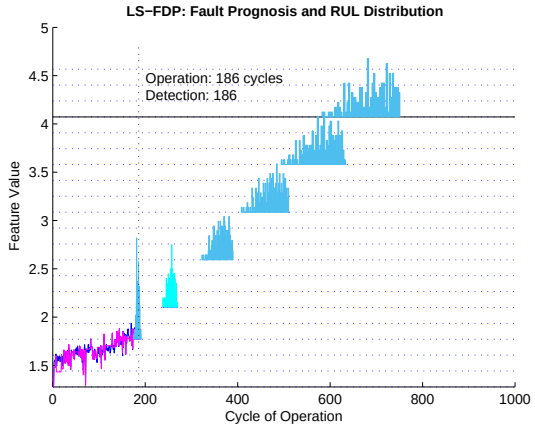


Figure 12. Experimental result of LS-based prognosis.

imental results. For the diagnosis, the two approaches show the comparable performance. In terms of prognosis, the LS-FDP shows better performance in terms of accuracy and precision. First, the introduction of Lebesgue sampling in FDP greatly reduce the computation time and the requirement of computation resources without sacrificing the performance of diagnosis. Since prognosis in Riemann sampling framework usually have a large prediction horizon, it often needs more computation time and resources. This in consequence becomes a main limitation of prognosis for those applications with fault tolerant control and reconfigurable control, where the real-time calculation of RUL is critical. Another important issue with large prediction horizon in Riemann sampling is the significant accumulation of uncertainties in prognosis and the degradation of the performance of prognosis in terms of accuracy and precision. The introduction of Lebesgue sampling in FDP provide a natural solution for real-time implementation, especially on those systems (such as embedded systems) with limited computation capability. The prediction horizon of LS-FDP can be very small comparing to that of RS-FDP, this is very good in managing the uncertainties in prognosis.

5. CONCLUSION AND FUTURE WORKS

This paper introduces a novel fault diagnosis and prognosis methodology that aims to: 1) introduce the concept of Lebesgue sampling into FDP and develop a novel FDP approach with an philosophy of "execution only when necessary" or an "as-needed" basis; and 2) enable the FDP on systems with limited computation capabilities, such as the embedded systems, that are widely used in automobiles, distributed diagnosis and prognosis, complex systems and networked systems. The methodology is composed of mathematically rigourous modules including the definition of diagnosis and prognosis in the framework of Lebesgue sampling with particle filtering. Other diagnostic and prognostic algorithms can be applied in this framework similarly. In the LS-FDP, diagnostic model and prognostic model need to be developed separately because fault diagnosis is based on the growth of fault dimension while prognosis is based on the calculation of operation time to reach different Lebesgue states defined as different fault dimensions. Experimental results from RS-FDP and LS-FDP on a planetary gear box with a seeded fault are presented and compared to illustrate the advantages of the proposed solution.

The use of Lebesgue sampling concept in fault diagnosis and prognosis are new in the research community of prognostic and health management. The paper only shows some preliminary results and there are many topics worth further research efforts. Some of the next step research include: 1) In this paper, the Lebesgue states are defined with uniform Lebesgue length. For some applications, the optimal Lebesgue length can be nonuniform and, therefore, the interval between Lebesgue states are not even. 2) Uncertainty management in prognosis is very important and critical. Although LS-FDP in many cases can reduce the prediction horizon and is naturally advantageous in uncertainty management, the theoretical and quantitatively analysis needs to be carried out to provide guidance for FDP algorithm design and implementation. There are many uncertainty management efforts in Riemann sampling based approaches and can be extended to LS-FDP approaches. 3) As we know, modeling is critical to the performance of FDP. The fault growth is a continuous process. For FDP, we discretize the model with Lebesgue sampling and therefore, it is necessary to investigate the accuracy loss caused by Lebesgue sampling. This result will provide a guidance for us on how to optimally choose the Lebesgue states and Exergue length. 4) For many applications, diagnosis and prognosis is not the goal but just the starting point for fault tolerance or system reconfiguration. It is of great interest to integrate the LS-FDP in Riemann sampling-based and Lebesgue sampling-based system reconfigurable control design.

REFERENCES

Agogino, A., Bonissone, P., Goebel, K., & Vachtsevanos, G. (2001). AI in equipment service. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *15*(4), 265-266.

Anger, C., Schrader, R., & Klingauf, U. (2012). Unscented Kalman filter with gaussian process degradation model for bearing fault prognosis. In *Proceedings of the european conference of the prognostics and health management society.*

Astrom, K., & Bernhardsson, B. (1999). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of ieee conference on decision and control.*

Balanban, E., & Slonso, J. (2013, Sept). A modeling frame-

work for prognostic decision making and its application to uav mission planning. In *Annual conference of the prognostics and health management society 2013.* New Orleans, LA.

Boskoski, P., & Urevc, A. (2011). Bearing fault detection with application to PHM data challenge. *International Journal of Prognostics and Health Management*, *2*(1), 1-10.

Celaya, J., Saxena, A., & Goebel, K. (2012). Uncertainty representation and interpretation in model-based prognostics algorithms based on Kalman filter estimation. In *Proceedings of the annual conference of the prognostics and health management society.* Minneapolis, MN.

Chen, C., Zhang, B., & Vachtsevanos, G. (2012). Prediction of machine health condition using neuro-fuzzy and Bayesian algorithms. *IEEE Transactions on Instrumentation and Measurement*, *61*(2), 297-306.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard, M. (2011). Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering. *IEEE Transactions on Industrial Electronics*, *58*(9), 4353-4364.

DeCastro, J., Tang, L., & Zhang, B. (2011). A safety verification approach to fault-tolerant aircraft supervisory control. In *Proceedings of aiaa guidance, navigation, and control conference.* Portland, OR.

de Freitas, N. (2002). Rao-Blackwellised particle filtering for fault diagnosis. In *Proceedings of the ieee aerospace conference* (Vol. 4, p. 1767-1772).

Edwards, D., Orchard, M., Tang, L., Goebel, K., & Vachtsevanos, G. (2010). Impact of input uncertainty on failure prognostic algorithms: Extending the remaining useful life of nonlinear systems. In *Proceedings of the annual conference of the prognostics and health management society.* Portland, OR.

Finkelstein, M. (2004). On the exponential formula for reliability. *IEEE Transactions on Reliability*, *53*(2), 265-268.

Goebel, K., Eklund, N., Hu, X., Avasarala, V., & Celaya, J. (2006). Defect classification of highly noisy NDE data using classifier ensembles. In *Smart structures and materials.* San Diego, CA.

Goebel, K., Saha, B., & Saxena, A. (2008, May). A comparison of three data-driven techniques for prognostics. In *Proceedings of the 62nd meeting of the society for machinery failure prevention technology.*

Hess, A., & Wells, S. (2003). Sliding mode control applied to reconfigurable flight control design. *Journal of Guidance, Control,and Dynamics*, *26*(3), 452-462.

Huang, W., & Dietrich, D. (2005). An alternavie degradation reliability modeling approach using maximum likelihood estimation. *IEEE Transactions on Reliability*, *54*(2), 310-317.

Isermann, R. (2005). Model-based fault detection and diagnosis-status and applications. *Annual Reviews in Control*, *29*, 71-85.

Jardine, A., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition based maintenance. *Mechanical Systems and Signal Processing*, *20*, 1483-1510.

Kaminskiy, M. (2005). A simple procedure for Bayesian estimation of the Weibull distribution. *IEEE Transactions on Reliability*, *54*(4), 612-616.

Lawless, J. (2003). *Statistical models and methods for lifetime data*. Hoboken: John Wiley and Sons.

Li, Y., Kurfess, T., & Liang, S. (2000). Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, *14*(5), 747-762.

McCann, R., & Le, A. (2008). Lebesgue sampling with a Kalman filter in wireless sensors for smart appliance networks. In *Proceedings of industry applciaitons society annual meeting.*

Morales-Menendez, R., de Freitas, N., Monterrey, I., Freitas, O. D., & Poole, D. (2002). Real-time monitoring of complex industrial processes with particle filters. In *Nips* (pp. 1433–1440).

Oppenheimer, C. H., & Loparo, K. A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. In *Aerosense* (pp. 122–132).

Orchard, M., Hevia-Koch, P., Zhang, B., & Tang, L. (2013, Nov). Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, *60*(11), 5260-5269.

Orchard, M., & Vachtsevanos, G. (2009). A particle filtering approach for online fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, *31*(3/4), 221-246.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 293-308.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On application the prognostic performance metrics. In *Proceedings of international conference on prognostics and health management.*

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium* (p. 107-114). LA USA.

Tang, L., Hettler, E., Zhang, B., & DeCastro, J. (2011). A testbed for real-time autonomous vehicle PHM and contingency management applications. In *Proceedings of international conference on prognostics and health management.* Montreal, Canada.

Tang, L., Zhang, B., DeCastro, J., & Hettler, E. (2011). An integrated health and contingency management case study on an autonomous ground robot. In *Proceedings*

of the 9th ieee international conference on control and automation.

Tumer, I., & Bajwa, A. (2004). A survey of aircraft engine health monitoring systems. In *Proceedings of the 35th aiaa/asme/ sae/asee joint propulsion conference* (p. 620-625).

Usynin, A., & Hines, J. (2007, November). Uncertainty management in shock models applied to prognostic problems. In *Aaai fall symposium.* Arlington, VA.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems.* John Wiley and Sons.

Yang, G. (2005). Accelated life test at higher usage rates. *IEEE Transactions on Reliability*, *54*(1), 53-57.

Zhang, B., Khawaja, T., Patrick, R., & Vachtsevanos, G. (2008). Blind deconvolution de-noising for helicopter vibration signals. *IEEE/ASME Transactions on Mechatronics*, *13*(5), 558-565.

Zhang, B., Khawaja, T., Patrick, R., & Vachtsevanos, G. (2010). A novel blind deconvolution de-noise scheme in failure prognosis. *Transactions of the Institute of Measurement and Control*, *32*(1), 3-30.

Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M., & Saxena, A. (2009). Application of blind deconvolution de-noising in failure prognosis. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 303-310.

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M., & Vachtsevanos, G. (2009). A probabilistic fault detection approach: application to bearing fault detection. *IEEE Transactions on Industrial Electronics*, *58*(5), 2011-2018.

Zhang, B., Tang, L., DeCastro, J., & Goebel, K. (2011, Aug). Prognostics-enhanced receding horizon mission planning for field autonomous vehicles. In *Aiaa guidance, navigation, and control conference.* Portland, OR.

Zhong, M., Fang, H., & Ye, H. (2007). Fault diagnosis of networked control system. *Annual Reviews in Control*, *31*(1), 55-68.

BIOGRAPHIES

Bin Zhang received the B.E. and M.E. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. Before that, he was with R&D, General Motors, Detroit, MI, USA, with Impact Technologies, Rochester, NY, USA, and with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests are prognostics and health management and intelligent systems.

Xiaofeng Wang received BS and MS degree in mathematics from East China Normal University in 2000 and 2003, respectively, and obtained his PhD degree in electrical engineering from the University of Notre Dame in 2009. After working as postdoctoral research associate in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, he joined the Department of Electrical Engineering as assistant professor at the University of South Carolina, Columbia, in 2012. His research interests include networked control systems, real-time systems, event-based control, robust adaptive control, distributed systems, and optimization.

Fault Detection and Severity Analysis of Servo Valves Using Recurrence Quantification Analysis

M. Samadani[1], C. A. Kitio Kwuimy[2], and C. Nataraj[3]

[1,2,3] *Department of Mechanical Engineering, Villanova University, Villanova, PA, 19085, USA*
msamadan@villanova.edu
cedrick.kwuimy@villanova.edu
nataraj@villanova.edu

ABSTRACT

This paper presents the application of recurrence plots (RPs) and recurrence quantification analysis (RQA) in model-based diagnostics of nonlinear systems. A detailed nonlinear mathematical model of a servo electro-hydraulic system has been used to demonstrate the procedure. Two faults have been considered associated with the servo valve including the increased friction between spool and sleeve and the degradation of the permanent magnet of the valve armature. The faults have been simulated in the system by the variation of the corresponding parameters in the model and the effect of these faults on the RPs and RQA parameters has been investigated. A regression-based artificial neural network has been finally developed and trained using the RQA parameters to estimate the original values of the faulty parameters and identify the severity of the faults in the system.

1. INTRODUCTION

Servo valves are complex electro-hydraulic systems which consist of very precise and sensitive components. A small change in the dimensions, metallurgical characteristics, or other parameters of these components can produce instability, error or even failure in the performance of the system. Hence, it is important to utilize effective algorithms and techniques to constantly monitor the performance of such systems and identify faults that can appear in them along with location and severity of the faults. Due to highly nonlinear characteristics of servo valves, it is essential to use techniques that can perform effectively in different domains of the nonlinear response.

In this paper, we introduce the application of recurrence plots (RPs) and recurrence quantification analysis (RQA) in model-based diagnostics of servo valves. The approach is general though and can be applied to any complex nonlinear system. Model-based fault detection approaches can be classified into three main categories of parity relation (Chow & Willsky, 1984; Gertler, 1997; Gertler & Singer, 1990), observer/filter-based (Frank & Ding, 1997; Patton, Frank, & Clarke, 1989) and parameter estimation (Isermann, 1982, 1984) methods. In parameter estimation method which is the main scope of this research, the parameters of the defective system are estimated and compared with the original parameters of the healthy system. The changes in parameter values are in many cases directly related to the defects. Therefore, this knowledge facilitates the fault diagnostics task. The parameter estimation technique has been used by many researchers for the detection of the faults in complex systems such as jet engines, rolling element bearings, DC motors, etc. (Baskiotis, Raymond, & Rault, 1979; Kappaganthu & Nataraj, 2011a; Liu, Zhang, Liu, & Yang, 2000). More information about parameter estimation based fault detection can be found in (Frank, Ding, & Koppen-Seliger, 2000; Isermann, 1997, 2005a, 2005b).

In general, nonlinear dynamic systems can exhibit diverse phenomena including multi-periodic, quasi-periodic and chaotic responses, as well as bifurcation and limit cycles. Many studies have reported the emergence of these complex nonlinear phenomena in industrial machinery originating from defects or due to their nonlinear nature (Sankaravelu, Noah, & Burger, 1994; Mevel & Guyader, 1993; Kappaganthu & Nataraj, 2011b). The prevailing parameter estimation methods are based on system identification techniques which are mostly suitable for linear systems and are not effective when the system response includes complex nonlinear phenomena. Moreover, the available methods require a pre-specified range for the initial guess of the parameter values which might not always be available in practice.

This paper presents the initial investigation of a new approach for parameter estimation-based diagnostics of nonlinear systems, based on the extracted information from the nonlinear response. Our main thesis is that the nonlinear dynamic

Mohsen Samadani et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

response of practical systems contains valuable information about the system including knowledge that could be used to develop an effective diagnostics framework. In an earlier work (Samadani, Kwuimy, & Nataraj, 2014, 2013) we presented an approach to extract information and features from the phase plane plot of the response in the periodic domain. The present paper extends that approach to systems with even more complex nonlinearities including quasi-periodicity using more advanced nonlinear dynamic analysis tools. The analysis in this paper is based on the recurrence properties of the system output in its reconstructed state space. In many cases, the phase space has dimensions higher than three which can only be visualized by projection into the two or three-dimensional sub-spaces. However, recurrence plots enable us to visualize and investigate certain aspects of the phase space trajectory in a two dimensional representation. The method of recurrence plots is a strong and effective tool for analysis of complex systems which has already been used for fault identification and diagnostics of nonlinear systems (Kwuimy, Samadani, Kappaganthu, & Nataraj, 2015). However, this is the first effort to use this method in a model-based approach to estimate the parameters of the system for fault diagnostics.

A detailed nonlinear mathematical model has been used to simulate the performance of the electro-hydraulic system. The analyses have been performed on the output flow of the servo valve. Three different electrical current signals including a periodic, a bi-periodic and a quasi-periodic signal have been input to the servo valve to investigate the performance of the algorithm in various nonlinear domains. RQA parameters have been obtained from the reconstructed phase space and used as the response features to identify dynamical changes in the system. Finally, an artificial neural network has been trained for mapping of the feature space to the parameter space.

The remaining parts of this paper are organized as follows. In section 2, a detailed mathematical model of the electro-hydraulic valve has been derived. In section 3, the definition of recurrence plots and RQA parameters have been provided. Section 4 describes the diagnostics algorithm along with the analyses and subsequent discussions. The conclusion is made in section 5.

2. MODELING OF THE ELECTRO-HYDRAULIC SERVO SYSTEM

A detailed dynamical model of a two-stage servo valve with a mechanical feedback has been used in the analyses. This system is shown in Fig. 1. Only the final equations are presented here. The detailed explanation of formulae can be found in (Samadani, Behbahani, & Nataraj, 2013; Rabie, 2009; Gordić, Babić, & Jovičić, 2004). The definition of system states and parameters along with the nominal values of the parameters have been presented in the nomenclature.

Neglecting the effect of the magnetic hysteresis, the net torque on the armature is given by the following expression.

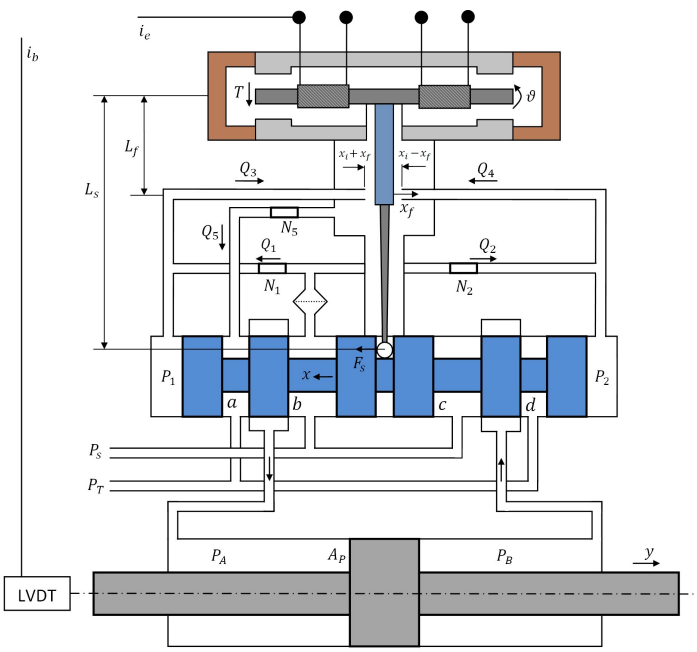


Figure 1. Functional schematic of the electro-hydraulic servo system [18]

$$T = K_i i_e \tag{1}$$

where the coefficient K_i can be calculated by:

$$K_i = \frac{N\lambda_p \mu_o A L}{2x_o^2} \tag{2}$$

The motion of the armature and the elements attached to it is described by the following equations:

$$T = J\frac{d^2\theta}{dt^2} + f_\theta \frac{d\theta}{dt} + K_T\theta + T_L + T_P + T_F \tag{3}$$

$$T_P = \frac{\pi}{4} d_f^2 (P_2 - P_1) L_f \tag{4}$$

The feedback torque depends on the displacement of the spool and the angle of the flapper and can be given by:

$$T_F = F_S L_S = K_S(L_S\theta + x)L_S \tag{5}$$

The rotational displacement of the flapper is limited mechanically by the jet nozzles. When the flapper reaches any of the side jet nozzles, a counter torque T_L is applied on it which can be calculated by the following equation:

$$T_L = \begin{cases} 0, & |x_f| < x_i \\ R_s \frac{d\theta}{dt} - (|x_f| - x_i) K_{Lf} L_f \text{sign}(x_f), & |x_f| > x_i \end{cases} \quad (6)$$

The flow rates through the flapper valve restrictions are given by the following equations:

$$Q_1 = C_D A_o \sqrt{\frac{2}{\rho}(P_s - P_1)} = C_{12}\sqrt{(P_s - P_1)} \quad (7)$$

$$Q_2 = C_D A_o \sqrt{\frac{2}{\rho}(P_s - P_2)} = C_{12}\sqrt{(P_s - P_2)} \quad (8)$$

$$\begin{aligned} Q_3 &= C_d \pi d_f (x_i + x_f)\sqrt{\tfrac{2}{\rho}(P_1 - P_3)} \\ &= C_{34}(x_i + x_f)\sqrt{(P_1 - P_3)} \end{aligned} \quad (9)$$

$$\begin{aligned} Q_4 &= C_d \pi d_f (x_i - x_f)\sqrt{\tfrac{2}{\rho}(P_2 - P_3)} \\ &= C_{34}(x_i - x_f)\sqrt{(P_2 - P_3)} \end{aligned} \quad (10)$$

$$x_f = L_f \theta \quad (11)$$

$$Q_5 = C_d A_s \sqrt{\frac{2}{\rho}(P_3 - P_T)} = C_5\sqrt{(P_3 - P_T)} \quad (12)$$

By using the continuity equation for the chambers of the flapper valve, the following expressions can be deduced:

$$Q_1 - Q_3 + A_s \frac{dx}{dt} = \frac{V_o - A_s x}{B}\frac{dP_1}{dt} \quad (13)$$

$$Q_2 - Q_4 - A_s \frac{dx}{dt} = \frac{V_o + A_s x}{B}\frac{dP_2}{dt} \quad (14)$$

$$Q_3 + Q_4 - Q_5 = \frac{V_3}{B}\frac{dP_3}{dt} \quad (15)$$

The motion of the spool is governed by the following equations.

$$A_s(P_2 - P_1) = m_s \frac{d^2x}{dt^2} + f_s \frac{dx}{dt} + F_j + F_s \quad (16)$$

$$F_j = \begin{cases} \left(\frac{\rho Q_b^2}{C_c A_b} + \frac{\rho Q_d^2}{C_c A_d}\right)\text{sign}(x) & \text{for } x > 0 \\ \left(\frac{\rho Q_a^2}{C_c A_a} + \frac{\rho Q_c^2}{C_c A_c}\right)\text{sign}(x) & \text{for } x < 0 \end{cases} \quad (17)$$

Ignoring the effect of transmission lines between the valve and the symmetrical hydraulic cylinder, the flow rates through the valve restriction areas are given by:

$$Q_a = C_d A_a(x)\sqrt{\frac{2}{\rho}(P_A - P_T)} \quad (18)$$

$$Q_b = C_d A_b(x)\sqrt{\frac{2}{\rho}(P_s - P_A)} \quad (19)$$

$$Q_c = C_d A_c(x)\sqrt{\frac{2}{\rho}(P_s - P_B)} \quad (20)$$

$$Q_d = C_d A_d(x)\sqrt{\frac{2}{\rho}(P_B - P_T)} \quad (21)$$

The area of the valve restrictions are given by:

$$\begin{cases} A_a = A_c = \omega c \\ A_b = A_d = \omega\sqrt{(x^2 + c^2)} \end{cases} \text{for } x \geq 0 \quad (22)$$

$$\begin{cases} A_a = A_c = \omega\sqrt{(x^2 + c^2)} \\ A_b = A_d = \omega c \end{cases} \text{for } x \leq 0 \quad (23)$$

Considering the internal leakage and neglecting the external leakage, the following equations can be obtained by applying the continuity equation to the cylinder chambers.

$$Q_b - Q_a - A_P\frac{dy}{dt} - \frac{(P_A - P_B)}{R_i} = \frac{(V_c + A_p y)}{B}\frac{dP_A}{dt} \quad (24)$$

$$Q_c - Q_a d + A_P\frac{dy}{dt} - \frac{(P_A - P_B)}{R_i} = \frac{(V_c - A_p y)}{B}\frac{dP_B}{dt} \quad (25)$$

Finally, the equation of motion for the cylinder piston is given by:

$$A_P(P_A - P_B) = m_p\frac{d^2y}{dt^2} + f_P\frac{dy}{dt} + K_b y \quad (26)$$

2.1. Servo Valve Faults

Various faults leading to parameter changes can appear in a servo valve. Three of the common defects in servo valves are:

- Change of magneto-motive force of the permanent magnet λ_p over time, which leads to the change of K_i
- Change of spool friction coefficient f_s, due to clearance variations or contamination

- Decrease in the diameter of nozzles d_f due to contamination or residuals

Sensitivity analyses show that the change of d_f does not significantly affect the dynamics of the system and hence, cannot be captured by dynamical analysis, unless the contamination blocks the nozzles completely. In this research, we assume the first two faults and use the response of the system in order to identify changes in those parameters. We suppose that one can measure the position of cylinder and the output flow of the valve.

3. RECURRENCE PLOTS AND RECURRENCE QUANTIFICATION ANALYSIS

The recurrence plots analysis for time series is based on the analysis of a matrix $\mathbf{R}$ whose elements are defined as:

$$\mathbf{R}_{ij} = \begin{cases} 1, & \mathbf{\Phi}_i \approx \mathbf{\Phi}_j, \\ 0, & \mathbf{\Phi}_i \neq \mathbf{\Phi}_j, \end{cases}, \quad i,j = 1...N, \tag{27}$$

where $\mathbf{\Phi}_i = (\phi_{1i}, \phi_{2i}, ..., \phi_{mi})$ is a state vector the dimension of m, N is the length of the time series, i and j are related respectively to the row and column of the matrix, and $\mathbf{\Phi}_i \approx \mathbf{\Phi}_j$ means equality up to an error ϵ.

If only a time series is available, the state vector $\mathbf{\Phi}$ can be reconstructed by using delay embedding theorem (Takens, 1981; Abarbanel, 1996; Fontaine, Dia, & Renner, 2011; Kwuimy, Samadani, & Nataraj, 2014). In this paper, the state vector has been reconstructed from the output flow of the valve. This is done in two steps: The first step consists of estimating the prescribed time lag T and the second step would be the evaluation of the embedded dimension m. In practice, if $u(i)$ is the available time series, the value of T corresponds to the first minimum of the average mutual information between the values of $u(i)$ and $u(i+T)$, and the embedding dimension can be deduced from the method of false nearness neighbor (Takens, 1981; Abarbanel, 1996; Kantz & Schreiber, 2004; Kwuimy et al., 2014). Once the values of T and m are obtained, the state vector $\mathbf{\Phi}$ can be reconstructed by:

$$\mathbf{\Phi} = (u(i), u(i+T), \ldots, u(i+T(m-1)) \tag{28}$$

The elements of the matrix $\mathbf{R}$ are thus obtained by comparing the state of the system at time i and j with a threshold precision ϵ. Thus, formally, one has:

$$\mathbf{R}_{ij} = \theta(\epsilon - ||\mathbf{\Phi}_i - \mathbf{\Phi}_j||), \tag{29}$$

with $||.||$ been the Euclidian norm (L_2-norm) and $\theta(y)$ is the heaviside function defined as:

$$\theta(y) = 1 \text{ for } y > 0 \text{ and } \theta(y) = 0 \text{ for } y < 0$$

Once we have the $\mathbf{R}$ matrix, the RP graph is obtained by plotting the R_{ij} points in the i and j plane with different colors. By definition, RP graphs are always symmetric ($R_{ij} = R_{ji}$) and always have a central diagonal.

In order to go beyond the qualitative impression given by RPs, complexity measures have been developed that quantify the structures of RPs and are called recurrence quantification analysis (RQA) (Zbilut, Thomasson, & Webber, 2002). In this paper, we use the following RQA parameters to quantify the RP of the system under various fault conditions.

- Recurrence rate (RR)

  The recurrence rate is the simplest RQA parameter which measures the density of recurrence points in a recurrence plot.

  $$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} \mathbf{R}_{i,j}(\epsilon) \tag{30}$$

- Determinism (DET)

  The determinism is the percentage of recurrence points which form diagonal lines in the recurrence plot of minimal length $\ell_{\min}$.

  $$DET = \frac{\sum_{\ell=\ell_{min}}^{N} \ell P(\ell)}{\sum_{\ell=1}^{N} \ell P(\ell)} \tag{31}$$

  where $P(\ell)$ is the frequency distribution of the lengths ℓ of the diagonal lines.

- Laminarity (LAM)

  In the same way, the amount of recurrence points forming vertical lines can be quantified by laminarity.

  $$LAM = \frac{\sum_{v=v_{min}}^{N} v P(v)}{\sum_{v=1}^{N} v P(v)} \tag{32}$$

  where $P(v)$ is the frequency distribution of the lengths v of the vertical lines, which have at least a length of $v_{\min}$.

- Average length of the diagonal lines (L)

  L is related with the predictability time of the dynamical system.

  $$L = \frac{\sum_{\ell=\ell_{min}}^{N} \ell P(\ell)}{\sum_{\ell=\ell_{min}}^{N} P(\ell)} \tag{33}$$

- Trapping Time (TT)

  The trapping time measures the average length of the vertical lines.

$$TT = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=v_{min}}^{N} P(v)} \tag{34}$$

- Entropy ($ENTR$)

  The probability that a diagonal line has exactly length ℓ can be estimated with $p(\ell) = \frac{P(\ell)}{\sum_{\ell=\ell_{\min}}^{N} P(\ell)}$. $ENTR$ is the Shannon entropy of this probability which reflects the complexity of the RP in respect of the diagonal lines.

$$ENTR = -\sum_{\ell=\ell_{min}}^{N} p(\ell)\ln p(\ell) \tag{35}$$

4. FAULT DIAGNOSTICS AND SEVERITY ANALYSIS

A standard procedure to identify faults and dynamical changes in systems is to input a pre-specified signal to the system, obtain the response and compare the signatures of the response with the ones of the system response in healthy conditions. Here we have input an electrical current signal to the servo valve, and measured the output flow of the valve. The state space of the system has then been reconstructed from the output flow signal and the effect of the parameter changes on the response has been evaluated using the defined recurrence quantification parameters.

In order to investigate the effectiveness of the approach in various domains of the nonlinear response, three different signals have been input to the servo valve including:

- Periodic input signal

$$i = 0.01\sin 50t$$

- Bi-periodic input signal

$$i = 0.01\sin 50t + 0.005\sin 75t$$

- Quasi-periodic input signal

$$i = 0.01\sin 50t + 0.005\sin 50\pi t$$

To better understand the effect of dynamical changes in recurrence point of view, the performance of the system is first analyzed and presented under three sample fault cases including:

- Healthy system
- Fault 1: K_i decreased by %50
- Fault 2: f_s increased by %500

Figure 2 shows the output flow of the valve versus time, corresponding to the three input cases, for the three sample fault scenarios.

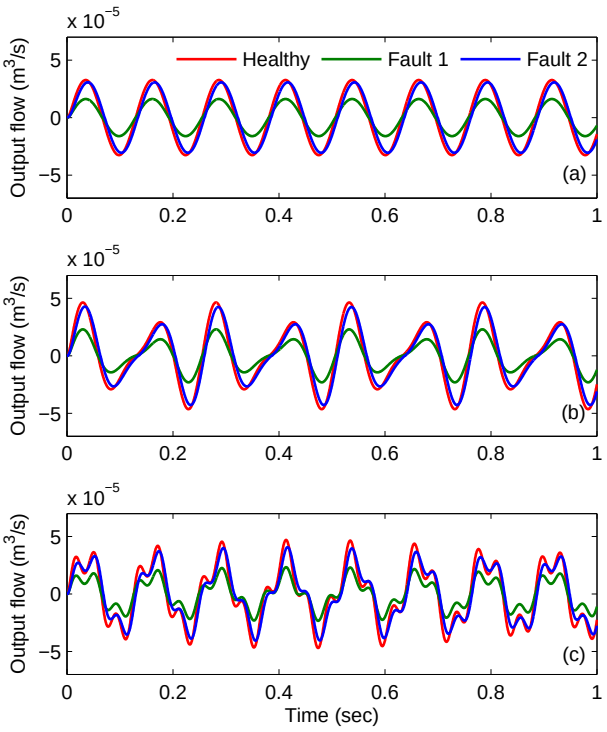


Figure 2. Time response of the system for (a): periodic, (b): bi-periodic and (c):quai-periodic inputs to the servo valve for three fault cases

In order to obtain the recurrence matrix and plots, we need to reconstruct the state space from the output flow time series. As discussed earlier, the appropriate time lag for the reconstruction of the state space corresponds to the first minimum in the average mutual information of the signal. Using this method, the time lag was determined to be T=50. By application of the method of false nearest neighbors, we found that the minimum embedding dimension for the system is d=2.

Figure 3 shows the recurrence plots of the reconstructed state space, for the three inputs and the three sample fault scenarios.

As can be seen, the plots consist of complicated patterns which are hard to interpret. In addition, there is little difference between them for the three fault cases, which is not easily detectable. Hence, we need quantitative measures to extract information from these plots.

Table 1 shows the computed RQA parameters for all nine cases. In this table p, bp, and qp correspond to the response of the system to the periodic, bi-periodic and quasi-periodic input signals, respectively. As can be seen, even though the difference of the recurrence plots for the three fault cases is hardly detectable by eye, RQA parameters can easily distinguish the differences and detect the alternations in the signal.

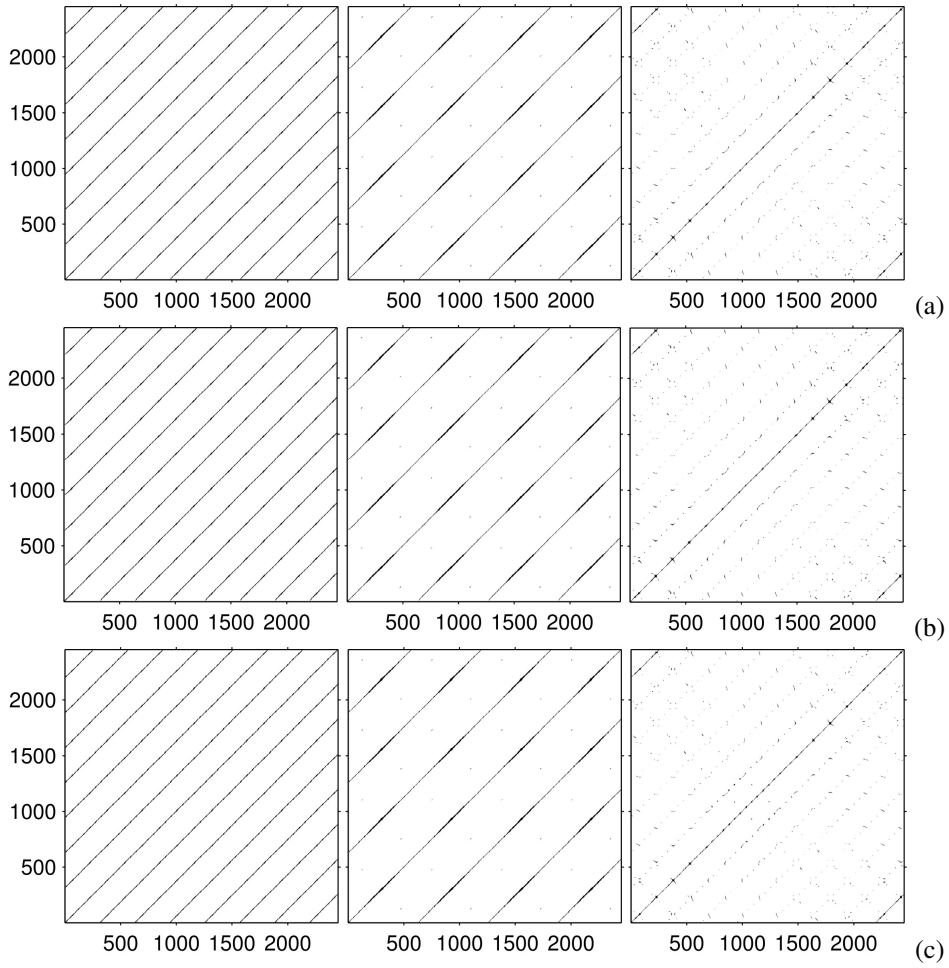


Figure 3. From left to right: Recurrence plots for periodic, bi-periodic and quasi-periodic inputs to the valve (a): Healthy system (b): Fault 1 (c): Fault 2

Table 1. RQA parameters for three defect cases

	Defect-free			Defect 1			Defect 2		
RQA Parameter	p	bp	qp	p	bp	qp	p	bp	qp
RR	0.0269	0.0160	0.0071	0.0269	0.0159	0.0070	0.0266	0.0160	0.0081
DET	0.9997	0.9994	0.9589	0.9997	0.9996	0.9589	0.9994	0.9996	0.9772
LAM	0.9999	0.9993	0.9548	0.9999	0.9993	0.9530	0.9997	0.9992	0.9790
L	107.1795	102.6667	5.6023	87.2343	95.2610	5.5580	103.0904	104.3878	6.7376
TT	7.4741	7.7383	4.4261	7.4741	7.7383	4.4261	7.4678	7.7575	4.9349
$ENTR$	3.7933	3.9199	1.9403	3.7933	3.9199	1.9403	3.8093	3.9572	2.2226

4.1. Mapping of Features to Parameters

So far we have illustrated how the response of the system is affected with the change of parameters and how it can be detected by using the RQA parameters. We were able to measure and represent these influences by quantitative criteria. In contrast to this, the diagnostics problem is the inverse problem, where we would like to predict the system parameters given its nonlinear response. In order to do that, machine learning techniques can be used which have been proved to be effective for diagnostics of machinery (Kankar, Sharma, & Harsha, 2011) and biomedical diagnostics (Jalali et al., 2014). In this paper, an artificial neural network (ANN) has been used. For this purpose, a two-layer feed-forward network with ten sigmoid hidden neurons and linear output neurons was developed. The inputs used for the training of the neural network were vectors of RQA parameters and the outputs were vectors of K_i and f_s. The data was obtained by random selection of the values of K_i and f_s in the intervals [0.1,0.6] and [1,100], respectively, simulation of the system and computation of the response features, i.e. RQA parameters, each time. A total number of 100 samples was used for training, validation and test of the network.

Figures 4, 5, and 6 show the regression plots of the network outputs with respect to targets for training, validation and test sets along the Regression (R) values for each case. For a perfect fit, the (R) value should be close to 1 and the data in the regression plot should fall along a 45 degree line, where the network outputs are equal to the targets. As can be seen, in this case, all the points have fallen along the 45 degree line and the R values are equal to 1, which are representatives of an accurate mapping of the features space to the parameters space.

Table 2 shows some samples of the performance of the parameter estimation systems developed with periodic, bi-periodic and quasi-periodic inputs. K_i^* and f_s^* represent the estimated values of K_i and f_s. This table shows that the proposed method has a very good ability to predict the original parameters of the system using the defined features, especially with the periodic input signal.

Table 2. Some examples of the performance of the parameter estimation system

		Periodic		Bi-periodic		Quasi-periodic	
K_i	f_s	K_i^*	f_s^*	K_i^*	f_s^*	K_i^*	f_s^*
0.1	5	0.098	5.879	0.096	6.087	0.123	6.088
0.3	50	0.289	50.623	0.275	51.025	0.356	51.610
0.6	100	0.591	101.511	0.592	98.410	0.633	102.214
0.2	25	0.207	24.234	0.206	25.324	0.227	26.665
0.4	2	0.390	2.012	0.384	2.622	0.383	3.001
0.5	10	0.512	9.824	0.488	9.357	0.520	9.512

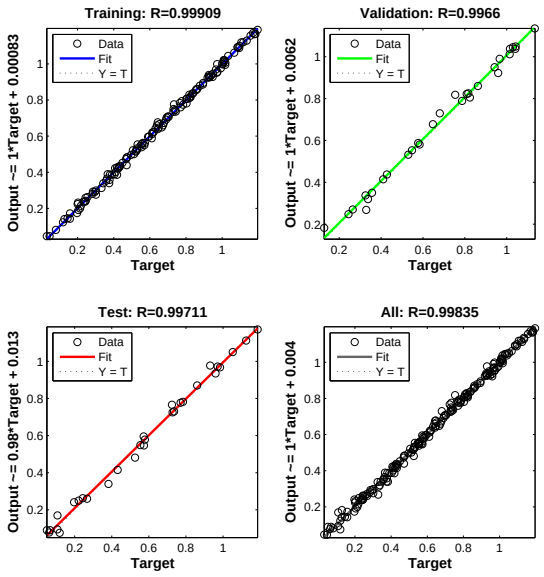


Figure 4. Outputs of the artificial neural network with respect to target values for the periodic input signal

5. CONCLUSION

We used recurrence plots and recurrence quantification analysis for model-based fault detection and diagnostics of an electro-hydraulic system. It was shown that the nonlinear response of the system contains valuable information about the system that can be used for this purpose. The analyses were performed with the assumption that only the output response of the system (here output flow of the valve) is available; and the other states were reconstructed using the method of time delays. The recurrence plots were produced and the corresponding recurrence analyses were performed on the reconstructed state space of the system. It was shown that even though the recurrence plots for the system with different faults can be similar, the dynamical changes can be detected by RQA parameters. An artificial neural network was trained using the RQA parameters to estimate the faulty parameters of the system. It was shown that RQA parameters can be used as effective features for characterizing the nonlinear response of the system even in the multi-periodic or quasi-periodic domain with complex nonlinearities.

In this study, the proposed method was only applied to numerical data obtained from the mathematical model of the system. Although the results were promising, there is no guarantee that we can obtain the same prediction accuracy for real experimental data. Hence, it is of importance to confirm the effectiveness of the approach with experimental analysis. In addition, only two parametric defects (defects due to change of parameter values) were considered in this paper, whereas in real world applications we might have multiple paramet-

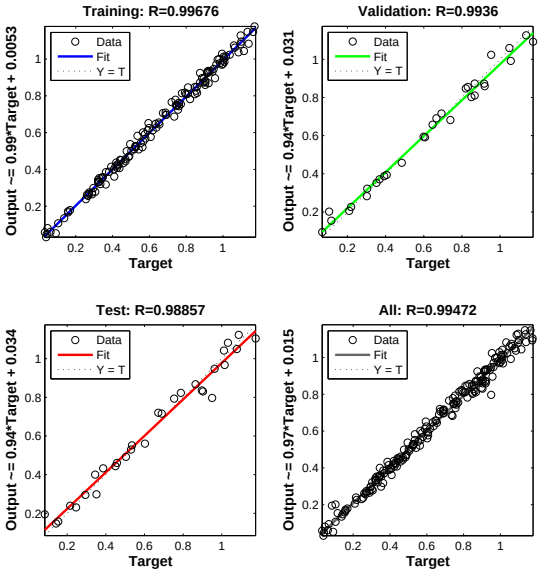


Figure 5. Outputs of the artificial neural network with respect to target values for the bi-periodic input signal

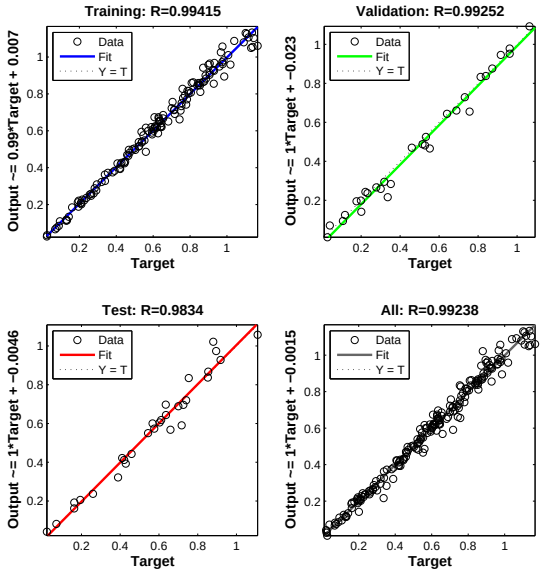


Figure 6. Outputs of the artificial neural network with respect to target values for the quasi-periodic input signal

ric defects in the system or even defects of the type that can change the structure of the mathematical model of the system. The present method can be extended with using more dynamical and statistical features in order to be able to characterize the system response and diagnose the faults in such conditions, which is currently the focus of our research.

ACKNOWLEDGMENT

This work is supported by the US Office of Naval Research under the grant ONR N00014-13-1-0485. We deeply appreciate this support. Thanks are due to Mr Anthony Seman III of ONR.

NOMENCLATURE

a	Width of spool edges	m	4e-03
A	Area of air gap	m^2	
A_5	Drain orifice area	m^2	
A_L	Area of the flow between spool and sleeve edges	m^2	
A_o	Orifice area	m^2	
$A_{a'}$, $A_{b'}$, $A_{c'}$, and $A_{d'}$	Spool valve restrictions areas	m^2	
A_P	Piston area	m^2	7e-04
A_s	Spool cross-sectional area	m^2	
b	Width of sleeve slots	m	4e-03
B	Bulk modulus of oil	Pa	1.5e09
c	Spool radial clearance	m	2e-06
C_c	Contraction coefficient		
C_d and C_D	Discharge coefficients		0.661
d_f	Flapper nozzle diameter	m	5e-04
d_5	Diameter of return orifice	m	6e-04
d_s	Spool diameter	m	4.6e-03
f_θ	Armature damping coefficient	Nms/rad	0.002
F_j	Hydraulic momentum force	N	
f_p	Piston friction coefficient	Ns/m	1000
f_s	Spool friction coefficient	Ns/m	3.05
F_s	Force acting at the extremity of the feedback spring	N	
H	Magneto-motive force per unit length	A/m	
i_b	Feedback current	A	
i_c	Control current	A	
i_e	Torque motor input current	A	
J	Moment of inertia of rotating part	Nms^2	5e-07
K_b	Load coefficient	N/m	0
K_{FB}	Feedback gain	A/m	1
K_{Lf}	Equivalent flapper seat stiffness	N/m	1e6
K_i	Current-torque gain	Nm/A	0.559
K_s	Stiffness of the feedback spring	N/m	900
K_T	Stiffness of flexure tube	Nm/rad	10.68
K	Rotational angle-torque gain	Nm/rad	9.45e-4
L	Armature length	m	0.029
L_f	Flapper length	m	0.009
L_s	Length of the feedback spring and flapper	m	0.03

L_{sp}	Length of spool land	m	1.5e-02
m_p	Piston mass	kg	5
m_s	Spool mass	kg	0.2
P_1	Pressure in the left side of the flapper valve	Pa	
P_2	Pressure in the right side of the flapper valve	Pa	
P_3	Pressure in the flapper valve return chamber	Pa	
P_A and P_B	Hydraulic cylinder pressures	Pa	
P_s	Supply pressure	Pa	1.2e7
P_T	Return line pressure	Pa	0
Q	Flow rate	m^3/s	
Q_1	Flow rate in the left orifice	m^3/s	
Q_2	Flow rate in the right orifice	m^3/s	
Q_3	Left flapper nozzle flow rate	m^3/s	
Q_4	Right flapper nozzle flow rate	m^3/s	
Q_5	Flapper valve drain flow rate	m^3/s	
Q_a, Q_b, Q_c, and Q_d	Flow rates through the spool valve restrictions	m^3/s	
R_i	Resistance to internal leakage	Ns/m^5	1e20
R_s	Flapper seat damping coefficient	$\mathrm{Nms}/\mathrm{rad}$	5000
T	Torque of electromagnetic torque motor	Nm	
T_F	Feedback torque	Nm	
T_L	Torque due to flapper displacement limiter	Nm	
T_P	Torque due to the pressure forces	Nm	
V_3	Volume of the flapper valve return chamber	m^3	5e-06
V_c	Half of the volume of oil filling the cylinder	m^3	1e-04
V_o	Initial volume of oil in the spool side chamber	m^3	2e-06
x	Spool displacement	m	
x_a	Displacement of the armature end	m	
x_f	Flapper displacement on the level of the jet nozzles	m	
x_i	Flapper displacement limit	m	3e-05
x_o	Length of the air gap in the neutral position of armature	m	3e-04
λ	Magneto-motive force	A	
λ_p	Magneto-motive force of the permanent magnet	A	66.75
μ	Permeability	Vs/Am	
μ_o	Permeability of the air	Vs/Am	4e-07
μ_r	Relative permeability		
ρ	Oil density	kg/m^3	867
ω	Width of ports on the valve sleeve	m	0.014
θ	Armature rotation angle	rad	

REFERENCES

Abarbanel, H. D. I. (1996). *Analysis of observed chaotic data*. Springer, New York.

Baskiotis, C., Raymond, J., & Rault, A. (1979). Parameter identification and discriminant analysis for jet engine machanical state diagnosis. In *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes* (Vol. 18, pp. 648–650).

Chow, E., & Willsky, A. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, *29*(7), 603–614.

Fontaine, S., Dia, S., & Renner, M. (2011). Nonlinear friction dynamics on fibrous materials, application to the characterization of surface quality. part i: global characterization of phase spaces. *Nonlinear Dynamics*, *66*(4), 625–646.

Frank, P., Ding, S. X., & Koppen-Seliger, B. (2000). Current developments in the theory of FDI. In *Proceedings of SAFEPROCESS* (Vol. 1, pp. 16–27).

Frank, P., & Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer–based fault detection systems. *Journal of Process Control*, *7*(6), 403–424.

Gertler, J. (1997). Fault detection and isolation using parity relations. *Control Engineering Practice*, *5*(5), 653–661.

Gertler, J., & Singer, D. (1990). A new structural framework for parity equation–based failure detection and isolation. *Automatica*, *26*(2), 381–388.

Gordić, D., Babić, M., & Jovičić, N. (2004). Modelling of spool position feedback servovalves. *International Journal of Fluid Power*, *5*(1), 37–51.

Isermann, R. (1982). Parameter adaptive control algorithms–a tutorial. *Automatica*, *18*(5), 513–528.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods–a survey. *Automatica*, *20*(4), 387–404.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods–an introduction. *Control Engineering Practice*, *5*(5), 639–652.

Isermann, R. (2005a). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer.

Isermann, R. (2005b). Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in Control*, *29*(1), 71–85.

Jalali, A., Buckley, E. M., Lynch, J. M., Schwab, P. J., Licht, D. J., & Nataraj, C. (2014, Jul). Prediction of periventricular leukomalacia occurrence in neonates after heart surgery. *IEEE Journal of Biomedical and Health Informatics*, *18*(4), 1453–1460.

Kankar, P., Sharma, S. C., & Harsha, S. (2011). Fault diagnosis of ball bearings using machine learning methods. *Expert Systems with Applications*, *38*(3), 1876–1886.

Kantz, H., & Schreiber, T. (2004). *Nonlinear time series analysis*. Cambridge University Press.

Kappaganthu, K., & Nataraj, C. (2011a). Mutual information based feature selection from data driven and model based techniques for fault detection in rolling element bearings. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 941–953).

Kappaganthu, K., & Nataraj, C. (2011b). Nonlinear modeling and analysis of a rolling element bearing with a clearance. *Communications in Nonlinear Science and Numerical Simulation*, *16*(10), 4134–4145.

Kwuimy, C. A. K., Samadani, M., Kappaganthu, K., & Nataraj, C. (2015). Sequential recurrence analysis of experimental time series of a rotor response with bearing outer race faults. In *Vibration engineering and technology of machinery* (pp. 683–696). Springer.

Kwuimy, C. A. K., Samadani, M., & Nataraj, C. (2014). Preliminary diagnostics of dynamic systems from time series. In *Proceedings of the ASME International Design Engineering Technical Conference.*

Liu, X.-Q., Zhang, H.-Y., Liu, J., & Yang, J. (2000). Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network. *IEEE Transactions on Industrial Electronics*, *47*(5), 1021–1030.

Mevel, B., & Guyader, J. (1993). Routes to chaos in ball bearings. *Journal of Sound and Vibration*, *162*(3), 471–487.

Patton, R. J., Frank, P. M., & Clarke, R. N. (1989). *Fault diagnosis in dynamic systems: theory and application*. Prentice–Hall, Inc.

Rabie, M. (2009). *Fluid power engineering*. McGraw Hill Professional.

Samadani, M., Behbahani, S., & Nataraj, C. (2013). A reliability-based manufacturing process planning method for the components of a complex mechatronic system. *Applied Mathematical Modelling*, *37*(24), 9829–9845.

Samadani, M., Kwuimy, C. A. K., & Nataraj, C. (2013). Diagnostics of a nonlinear pendulum using computational intelligence. In *ASME 2013 Dynamic Systems and Control Conference.*

Samadani, M., Kwuimy, C. A. K., & Nataraj, C. (2014). Model-based fault diagnostics of nonlinear systems using the features of the phase space response. *Communications in Nonlinear Science and Numerical Simulation*.

Sankaravelu, A., Noah, S. T., & Burger, C. P. (1994). Bifurcation and chaos in ball bearings. *ASME Applied Mechanics Division–Publications–AMD*, *192*, 313–313.

Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence* (pp. 366–381). Springer.

Zbilut, J. P., Thomasson, N., & Webber, C. L. (2002). Recurrence quantification analysis as a tool for nonlinear exploration of nonstationary cardiac signals. *Medical Engineering & Physics*, *24*(1), 53–60.

BIOGRAPHIES

Mohsen Samadani Mohsen received his B.Sc and M.Sc in Mechanical Engineering from Isfahan University of Technology, Isfahan, Iran. He is currently a Ph.D. candidate at the Department of Mechanical Engineering at Villanova University. Mohsen has been involved in various research topics including manufacturing technologies, control, vibrations, system dynamics, hydraulic systems and reliability analysis. His current research interests include data analysis, machine learning, nonlinear dynamics and vibrations with applications to machinery diagnostics and health management. He is a member of Sigma Xi and ASME and a recipient of Sigma Xi best poster award and PHM doctoral consortium travel award.

Cedrick Kwuimy Prior to joining the Department of Mechanical Engineering at Villanova University in Jan. 2011, Dr. Kwuimy worked in South Africa as a postdoctoral research associate at the African Institute for Mathematical Sciences (2009-2010) and as a Research and Teaching Assistant (2007-2009) at the Faculty of Science at the University of Yaounde, Cameroon. He has been involved in a wide range of research topics including vibration control, nonlinear dynamics of self-sustained electromechanical devices, synchronization, nonlinear analysis of butterfly valves, and chaos control and prediction in active magnetic bearings. He has over 30 peer-reviewed papers in international journals and conference proceedings and serves as a reviewer in high standard journals including Nonlinear Dynamics, Journal of Vibration and Control and Journal of Sound and Vibration. Dr. Kwuimy has supervised three graduate research theses at the African Institute for Mathematical Sciences and is the recipient of Victor Rothschild Fellowships at African Institute for Mathematical Sciences and Research.

C. Nataraj Dr. C. Nataraj holds the Mr. and Mrs. Robert F. Mortiz, Sr. Endowed Chair Professorship in Engineered Systems at Villanova University. He has a B.S. in Mechanical Engineering from Indian Institute of Technology, and M.S. and Ph.D. in Engineering Science from Arizona State University. After getting his Ph.D. in 1987, he worked for a year as a research engineer and a partner with Trumpler Associates, Inc. He is currently the Chairman of the Mechanical Engineering Department at Villanova University. Dr. Nataraj was also the founding director of the Center for Nonlinear Dynamics and Control in the College Of Engineering. He has worked on various research problems in nonlinear dynamic systems with applications to mobile robotics, unmanned vehicles, rotor dynamics, vibration, control, and electromagnetic bearings. His research has been funded by Office of Naval Research, National Science Foundation, National Institute of Health and many companies.

Model-Based Fault Diagnosis of a Planetary Gear Using Transmission Error

Jungho Park[1], Jong Moon Ha[2], Byeng D. Youn[3], Sang Hyuck Leem[4], Joo-Ho Choi[5], and Nam Ho Kim[6]

[1, 2, 3]*Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, 151-742, Republic of Korea*
hihijung@snu.ac.kr
billlyhjm@snu.ac.kr
bdyoun@snu.ac.kr

[4,5]*Department of Aerospace & Mechanical Engineering, Korea Aerospace University, Goyang, Gyeonggido, 412-791, Republic of Korea*
sanghuyck@naver.com
jhchoi@kau.ac.kr

[6]*Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA*
nkim@ufl.edu

ABSTRACT

A Planetary gear can transmit high torque ratio stably and, therefore, the gear is widely used in industrial applications, i.e., wind turbines, automobiles, helicopters. Unexpected failure of the planetary gear results in substantial economic loss and human casualties. Extensive efforts have been made to develop the fault diagnostic techniques of gears; however, the techniques are mostly concerned about spur gears. This is mainly because understanding of complex dynamic behaviors of a planetary gear is lacking, such as multiple gear contacts, non-stationary axis of rotation, etc. This study thus proposes model-based fault diagnostics for a planetary gear that is based upon its dynamic analysis. Instead of vibration signals, this study uses transmission error (TE) signals for fault diagnostics of the planetary gear because TE signals (a) are directly related to the dynamic behaviors of gear mesh stiffness and (b) increase as damages on a gear mesh reduce the gear mesh stiffness. A lumped parameter model was used for modeling dynamic behaviors of the planetary gear. For more precise modeling, mesh phase difference–between sun, ring, and planet gear– and contact ratio were taken into account in the lumped parameter model. After acquiring transmission error signals from the model, order analysis and data processing were executed to generate health related data for the planetary gear. Consequently, it is concluded that the use of transmission error signals helps gain understanding of complex dynamic behaviors of the planetary gear and diagnose its potential faults.

Jungho Park et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

A planetary gear is a kind of gear system composed of a ring gear, sun gear, planet gear and carrier as shown in Figure 1. While the ring gear is covering the whole gearbox, multiple planet gears connected by a carrier are rotating around the sun gear. As planet gears are distributing the loads a gear system delivers, the planetary gear can transmit high torque ratio in a stable way. So it is commonly used in many huge engineering applications like wind turbines, automobiles, helicopters. As unexpected failure of the planetary gear can result in substantial economic loss and human casualties, fault diagnostics for various gear system including the planetary gear has been developed.

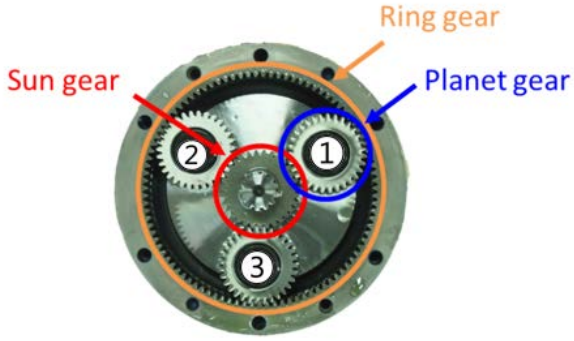


Figure 1. Cross-sectional view of a planetary gear.

Zheng, Li, and Chen (2002) developed fault diagnostics of a spur gear based on continuous wavelet transform. Samanta (2004) presented a comparative study for the performance of fault diagnostics for a spur gear between artificial neural networks (ANNs) and support vector machines (SVMs) which classify the normal and fault condition. Saravanan, Cholairajan, and Ramachandran (2009) used fuzzy classifier with vibration signal to detect the fault of a spur bevel gear box. Fault diagnostics for a planetary gear is relatively less developed (Lei, Kong, Lin, and Zuo, 2012). Barszcz and Randall (2009) applied spectral kurtosis technique to detect a tooth crack of the planetary gear. Lei et al. (2012) proposed two new diagnostics parameters for the planetary gear, root mean square of the filtered signal (FRMS) and normalized summation of positive amplitudes of the difference spectrum between the unknown signal and the healthy signal (NSDS). Feng and Liang (2014) exploited the adaptive optimal kernel (AOK) method to deal with the non-stationary signal of the planetary gear. Above literatures used vibration signals to detect the faults of the gear system. In recent years, Acoustic emission signals has been used to detect the faults of a gear due to the sensitivity to early faults than vibration signal. Qu, He, Yoon, Van Hecke, Bechhoefer, and Zhu (2014) performed comparative study between vibration signal and acoustic emission signal. They found that acoustic emission signal is more sensitive to small tooth damage in the low speed range.

However, previous signals used for fault diagnostics of gears have defects because they didn't utilize the physical meaning of gear dynamics. Gear system is a very well organized system, especially a planetary gear has its own peculiarity due to the gear dynamics arising from pitch, contact ratio, phase difference. Therefore, we introduced new fault diagnostics signal, Transmission Error (TE) in a lumped parameter model. TE is defined as "the angular difference between the position that the output shaft of a gear drive would have if the gearbox were perfect (without errors or deflections) and the actual position of the output shaft" according to Remond and Mahfoudh (2005). This signal is deeply related with gear mesh stiffness. So, it has physical meaning in gear dynamics and could have potentials which could classify the fault condition in gear system. In this paper, we compared the TE signal from simulation model in both normal and faulty planetary gear and demonstrated the validity of TE for fault diagnostics of a planetary gear.

This paper is organized as follows. The development of the planetary gear lumped model is described in Section 2. In section 3, Description about how TE could have physical meaning and relation with fault is followed. Section 4 presents the way we processed the signal to effectively observe the fault symptom and results are shown. In section 5, health indices used for fault diagnostics of a planetary gear are introduced and they are calculated from TE signal for normal and faulty gear obtained from simulation model. Finally, section 6 states the conclusion and future work of this research.

2. PLANETARY GEAR MODELING

A Planetary gear used in this paper is constructed using DAFUL 4.2. Basic lumped parameter modeling strategies for planetary gears in DAFUL 4.2 are based on a thesis from Kim (2001).

2.1. Basic Specification of a Planetary Gear

Basic gear specification used in this paper is as shown in Table 1. These parameters are used as input parameters for lumped parameter model. For example, numbers of teeth for each gear are used for calculating the gear ratio (4.06:1), and pressure angle information is used for indicating the direction of interacting force, and so on. The system input is a low speed shaft connected with a carrier and the system output is a high speed shaft connected with a sun gear.

Table 1. Planetary gear specification.

Gear data	Sun	Ring	Planet
Number of teeth	31	95	31
Pressure angle (deg)	20	20	20
Module (mm)	1.5	1.5	1.5
Pitch circle diameter (mm)	46.5	46.5	142.5
Dedendum circle diameter (mm)	43.643	146.25	43.409
Tip diameter (mm)	50.693	139.5	50.459
Whole depth (mm)	3.525	3.375	3.525
Face width (mm)	16	16	16

2.2. Gear Mesh Stiffness

Another important parameter used in DAFUL is gear mesh stiffness. Gear mesh stiffness is defined as the ratio between the input torsional load and the total angular rotation of the gear (Sirichai, Howard, Morgan, and The, 1997). As mesh stiffness is closely related to the TE, which we would use as a fault signal, it is carefully parameterized in DAFUL. In DAFUL, gear mesh stiffness can be parameterized based on (a) one mesh, (b) all mesh, or (c) constant value.

2.2.1. Magnitude of Gear Mesh Stiffness

The magnitude of gear mesh stiffness has repeating patterns due to the repeating contact condition (single, double contact) in path of contact of gear mesh. This gear mesh stiffness can be obtained analytically (Cornell, 1985). However, in this paper, it is calculated by finite element analysis code, ABAQUS, and the result is as Figure 2. The Figure 2 is for ring-planet gear mesh stiffness calculated from ABAQUS code. Then, this values were parameterized as two values,

448900 and 536700N/mm for simplicity. Sun-planet gear mesh stiffness is achieved in the similar way and they were also parameterized as two values, 210600 and 274000N/mm.

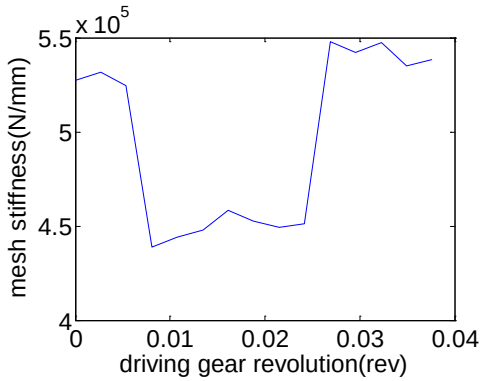


Figure 2. Ring-planet gear mesh stiffness result from finite element analysis.

We can also notice that ring-planet gear mesh stiffness is bigger than sun-planet gear mesh stiffness about two times. This is because ring-planet gear is an internal gear which shows high contact ratio.

2.2.2. Phase of Gear Mesh Stiffness

Parker and Lin (2003) calculated phase difference of gear mesh stiffness not only among planets with a ring gear and sun gear but also between ring-planet gear mesh stiffness and sun-planet gear mesh stiffness in a planetary gear.

For the case of gear mesh stiffness among planets with a ring gear and sun gear, the phase difference can be calculated by the following equation when the planet rotation is counter-clockwise.

$$\gamma_{sn} = -\frac{Z_s \psi_n}{2\pi} \quad \gamma_{rn} = \frac{Z_r \psi_n}{2\pi} \tag{1}$$

where γ_{sn} is relative phase difference between *n*th sun-planet gear mesh stiffness and the reference sun-planet gear mesh stiffness, γ_{rn} is relative phase difference between *n*th ring-planet gear mesh stiffness and the reference ring-planet gear mesh stiffness, $Z_{r,s}$ is ring and sun gear tooth numbers and ψ_n is circumferential angle measured at reference planet gear. In this equation reference planet gear can be selected arbitrarily as 1st planet gear in Figure 1. For our case, Z_r, Z_s are 95, 31 and $\psi_{1,2,3}$ are 0, $2\pi/3$, $4\pi/3$ respectively. So $\gamma_{r1}, \gamma_{r2}, \gamma_{r3}$ are 0, 2/3, 1/3 and $\gamma_{s1}, \gamma_{s2}, \gamma_{s3}$ are 0, -1/3, -2/3 respectively, which means same phase difference to the same planet with a ring gear and sun gear as phase difference of 2/3, 1/3 is identical to phase difference of -1/3, -2/3.

For the case of gear mesh stiffness between ring-planet gear mesh stiffness and sun-planet gear mesh stiffness in a planetary gear, the phase difference (= γ_{rs}) can be calculated analytically based on pitch contact point which is the midpoint of the lower stiffness region. It is indicated as a red

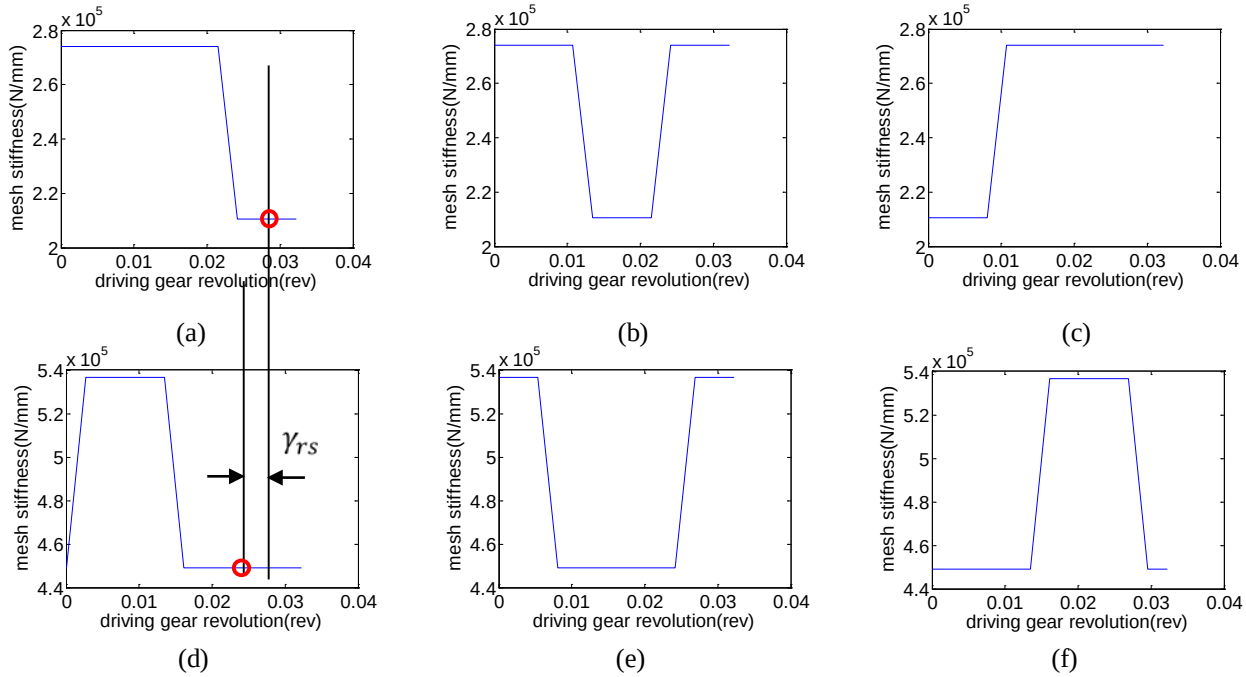


Figure 3. Gear mesh stiffness of (a) 1st sun-planet gear, (b) 2nd sun-planet gear, (c) 3rd sun-planet gear, (d) 1st ring-planet gear, (e) 2nd ring-planet gear, (f) 3rd ring-planet gear.

circle in Figure 3 and it is applied to in the phase of a planetary gear.

It is proved that no phase difference in sun-planet gear mesh stiffness could make equal load distribution at planets, and differing phase difference could have significant effect in reducing vibration and noise. (Parker & Lin, 2003) In our case, as phase difference is equally distributed at each planet, we could guess our planetary gear is designed to reduce the noise and vibration rather than to distribute the loads the system carries.

2.3. Mesh Stiffness of a Faulty Gear

In this paper, we define gear fault as a crack in a planet gear tooth. Chaari and Haddar (2009) studied the relationship between crack size and mesh stiffness reduction. In above literature, gear mesh stiffness for a spur gear gets smaller and smaller as a crack in a gear tooth gets larger. And this literature showed that 1/4 of tooth thickness- cracked gear induces 10% mesh stiffness reduction to the one of whole gear mesh stiffness. In this research, therefore, as each ring-planet gear and sun-planet gear interaction can be thought as a spur gear interaction, mesh stiffness reduction would happen to the both ring-planet and sun-planet gear mesh stiffness by 10% in the same way if we assume a crack in a gear tooth is 1/4 of tooth thickness. However, in the planetary gear, we should also consider the fault phase difference between ring-planet gear mesh stiffness reduction and sun-planet gear mesh stiffness reduction. The planet gear makes one rotation around a sun gear while it is meshing with a ring gear and sun gear repeatedly. So for the crack in a gear tooth, it contact with a sun gear, ring gear, sun gear at 0, 1/2, 1 rotation of planet gear like Figure 4. So, the mesh stiffness phase difference in fault condition is 1/2 rotation of a planet gear like Figure 5. So this stiffness values were parameterized for the faulty gear mesh stiffness of the planetary gear.

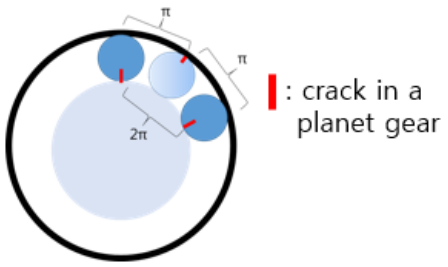


Figure 4. A cracked planet gear rotation behavior.

3. SIGNALS FOR FAULT DETECTION

This section will discuss TE, the signal used for fault detection in this research. First, we explain about why TE is related with health condition and how TE varies when the fault is seeded into gear sets. And then, TE behavior of a planetary gear in normal condition will be discussed.

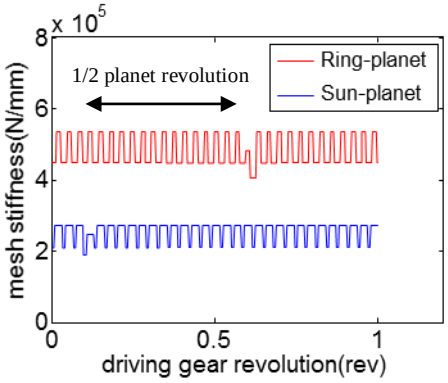


Figure 5. Ring-planet and sun-planet gear mesh stiffness for a cracked planetary gear.

3.1. Transmission Error

TE can be simply defined as "the output gear difference between the expectation and reality". TE occurs due to many sources like tooth profile error, tip relief error, mesh stiffness, etc. In our case, we only consider the effect from mesh stiffness. Let's say the gear is rotating clockwise and inverse torque is applied to output gear counterclockwise. Then the gear teeth will deflect counterclockwise due to inverse torque. This is the reason TE happens in a gear. That is, for the single contact condition, gear mesh stiffness is low, and TE would show higher value. Then, for the double contact condition, gear mesh stiffness is high, and TE would show lower value. In this way, TE fluctuates repeatedly along the stiffness fluctuation. Then, what would happen if a gear tooth is cracked? As we discussed in section 2, crack in a gear tooth makes gear mesh stiffness reduction. So, TE would increase as the stiffness is reduced. In this way, TE signal can be a physically meaningful signal differently from other signals in relation with mesh stiffness. Also, as stiffness reduces gradually along the crack size propagation, TE signal can be a more useful signal for fault prognostics.

3.2. Transmission Error in a Planetary Gear

Transmission error in a planetary gear can be calculated as

$$TE = h.s.s\ rotation - gear\ ratio \times l.s.s\ rotation \quad (2)$$

where *h.s.s* denotes high speed shaft connected with a sun gear and *l.s.s* denotes low speed shaft connected with a carrier. Differently from a spur gear, TE in a planetary gear shows complicated behavior due to the effect from multiple meshing condition from ring, planet, and sun gear as in Figure 1. Figure 6 shows a TE signal result from DAFUL when input velocity is 20rad/s, inverse torque is 2×106 Nmm with sampling frequency 1000hz. 3 peaks in one fluctuation are

repeatedly appearing. We could guess this could happen due to the effect from three planets.

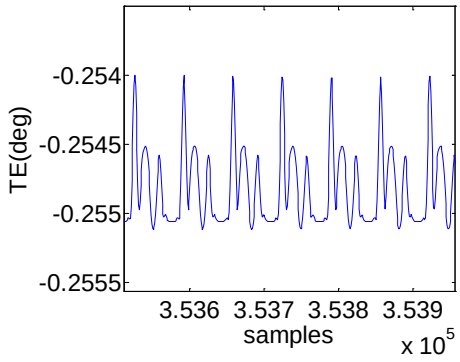


Figure 6. Simulated transmission error signal from a planetary gear.

4. SIGNAL PROCESSING

To effectively observe the fault characteristics of a planetary gear, TE signals were processed with three steps like Figure 7, (1) DC component subtraction, (2) Time synchronous averaging (TSA), (3) Order analysis. In this section, we discuss the principles of each procedure for signal processing and explain why each procedure was performed.

4.1. DC Component Subtraction

The first step for signal processing is to subtract DC component in raw TE signal. TE fluctuates while the DC component is shifted due to the deflection like Figure 6. To effectively analyze the TE in a frequency domain, mean value of the TE should be subtracted from original signal. Comparing with Figure 7 (a) and 7 (b), you can see TE value is shifted along y-axis.

4.2. Time Synchronous Averaging (TSA)

Time synchronous averaging (TSA) for gear signal analysis was originally proposed to suppress the noisy signal - (a) non-synchronous coherent signal, (b) non-coherent random signal (Hochmann & Sadok, 2004.). However, in this research, TSA was adopted to effectively observe the gear mesh frequency of interest in TE signal. Eq. (3) is the equation used for TSA in this paper.

$$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{3}$$

where $\bar{x}$ is time synchronous averaged data, N is number of planet rotation and x_k is TE data in time domain for kth planet rotation.

By calculating equation (3), we can observe the only planet-oriented behavior of TE signal. In Figure 7 (c), there are 31 fluctuations which contain 3 peaks in a fluctuation as in Figure 6. 31 is the number of a planet gear and we can observe how the TE is varying for the 1 rotation of a planet gear by performing TSA.

4.3. Order Analysis

Then the order analysis was performed to analyze the effect from the planet gear mesh frequency. This can be performed by transforming time-domain TE data into frequency domain by Fast fourier transform (FFT) code in MATLAB. As TSA data were averaged with planet rotation, we can observe the planet gear tooth number component and its harmonic in order analysis result in Figure 7 (d).

4.4. Results from Normal and Cracked Gear

After following these procedures, simulated TE results from normal and cracked planetary gear were obtained like Figure 8, 9. Figure 8 shows the TSA of TE from normal and cracked planetary gear. In advance, we can see the two sparks in Figure 8 (b). In Figure 4, we showed that a crack in a planet gear contacts with a ring, sun gear repeatedly while planet gear makes one rotation. So, this behavior makes TE in a planetary gear spark from normal TE. Also there is magnitude difference in TE sparks. As there are difference in stiffness between ring-planet and sun-planet gear mesh stiffness, TE sparks, which arose from stiffness, also has difference in magnitude.

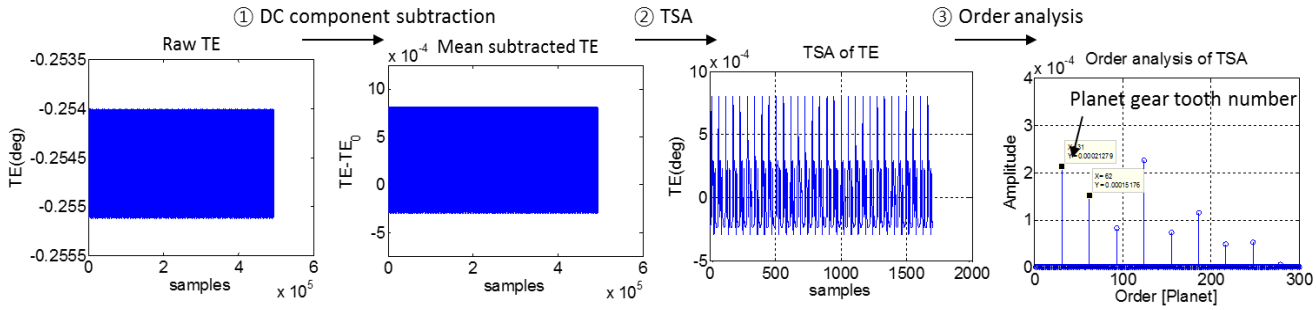
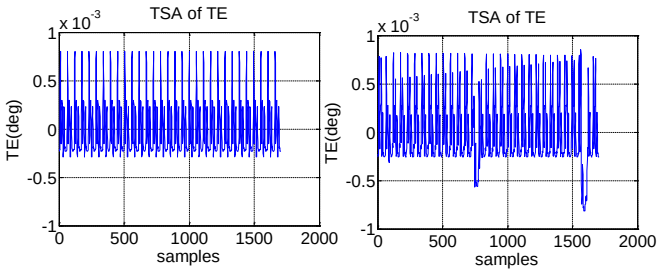


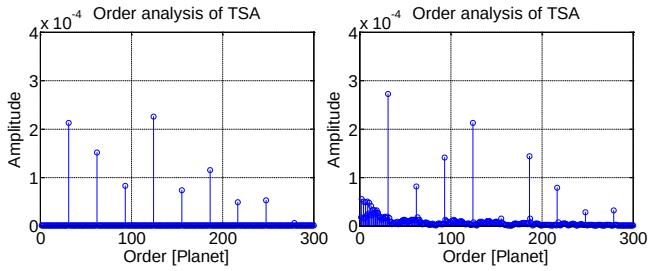
Figure 7. Procedures for TE signal processing.

Figure 9 shows the order analysis results. In Figure 9 (b), we can observe the sub-harmonic and sideband near the main harmonic.



(a) (b)

Figure 8. TSA results of TE in a (a) normal and (b) cracked planetary gear.



(a) (b)

Figure 9. Order analysis results of TE in a (a) normal and (b) cracked planetary gear.

5. HEALTH INDEX CALCULATION

TE results from section 4.4 need to be quantified to properly represent health state of the system. Lebold, McClintic, Campbell, Byington, and Maynard (2000) organized health index frequently used for gearbox diagnostics. In section 5, we adopted two health index and compared the results from normal and cracked gear.

5.1. Health Index

In this study, we adopted root mean square (RMS) and FRMS to quantitatively classify a cracked gear from a normal gear.

First, RMS can be formulated as

$$RMS = \sqrt{\frac{1}{N}\sum_{k=1}^{N} x_k^2} \tag{4}$$

where x_k is kth time data point and N is number of total data. By calculating RMS, overall noise level can be easily detected.

Secondly, FRMS can be formulated as

$$FRMS = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (s(t))^2} \tag{5}$$

where *s(t)* is the ith data of data point of the filtered signal S and T is the number of total data. Filtered signal is obtained by filtering out the shaft frequency and its five-order harmonics and gear mesh frequency and its three-order harmonics in frequency domain. Then the signal is transformed into time domain again. This signal is effective in planetary gear analysis because shaft frequency and its harmonics, gear mesh frequency and its harmonics mainly dominates the vibration signal of planetary gear (Yaguo, et al., 2012).

5.2. Health Index from Various Condition

To verify the validity of the TE as a fault diagnostics signal, health indices proposed from section 5.1 are calculated using TE in various conditions.

First, RMS, FRMS were calculated from various input speed at 1~20 rad/s like Figure 10. Then, RMS, FRMS were calculated from various inverse torque at $1\sim10\times10^5$ Nmm magnitude like Figure 11. We can observe that at faster input speed and higher inverse torque magnitude we can more easily differentiate the cracked gear from a normal gear.

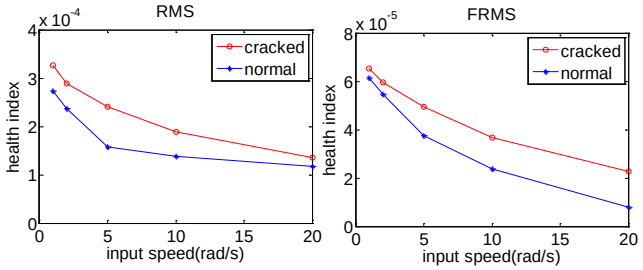


Figure 10. RMS, FRMS values from a normal, cracked planetary gear at various input speed.

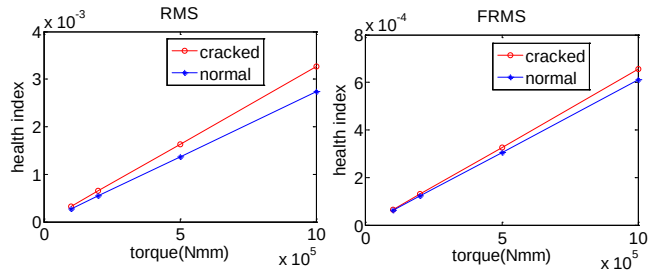


Figure 11. RMS, FRMS values from a normal, cracked planetary gear at various inverse torque.

Also, we observed the RMS and FRMS change along the relative stiffness like Figure 12. Relative stiffness means the ratio of the stiffness to the stiffness from normal planetary

gear. As bigger crack size indicates larger gear mesh stiffness reduction, we tried to estimate health index from different crack size from relative stiffness. From Figure 12, we can notice that as crack size is getting bigger, health indices indicates larger values.

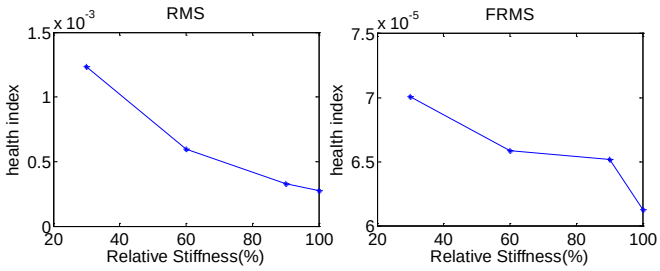


Figure 12. RMS, FRMS values from different crack size.

6. CONCLUSION

This paper proposed a new signal, TE for model-based fault diagnostics of the planetary gear. First, we developed a planetary gear with lumped parameter model. In this step, we closely studied phase difference in ring-planet gear mesh stiffness and sun-planet gear mesh stiffness considering pitch contact point. To simulate the fault condition in a gear as a crack in a gear tooth, we studied the relationship between crack size and gear mesh stiffness, which is directly related with TE signal. We also considered the fault phase occurring from planet gear rotation. Then we analyzed the TE signal in an organized signal processing procedures and calculated health indices. By calculating health indices from various condition, we could conclude that TE can be a good signal for diagnosing the fault in a planetary gear. Moreover, as TE is a physically meaningful signal related with stiffness, it can not only differentiate fault level but also be a signal for fault prognosis.

Future work will include development of lumped parameter model and validation using test-bed data. As we considered many things in modeling the planetary gear, it can be developed more precisely to simulate a real planetary gear. Then, finally, validation using a real planetary gear TE data should be performed. To accurately measure the TE signal, many methods have been developed using encoder. So, by obtaining and analyzing the TE data, proposed idea could be validated.

ACKNOWLEDGEMENT

This work was supported by the International Collaborative Energy Technology R&D Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea. (No. 20118520020010)

REFERENCES

Barszcz, T., & Randall, R. B. (2009). Application of spectral kurtosis for detection of a tooth crack in the planetary gear of a wind turbine. *Mechanical Systems and Signal Processing*, 23(4), 1352-1365.

Chaari, F., Fakhfakh, T., & Haddar, M. (2009). Analytical modelling of spur gear tooth crack and influence on gearmesh stiffness. *European Journal of Mechanics-A/Solids*, 28(3), 461-468.

DAFUL (2013). DAFUL User's Manual Ver. 4.2, Virtual Motion, Inc.

Feng, Z., & Liang, M. (2014). Fault diagnosis of wind turbine planetary gearbox under nonstationary conditions via adaptive optimal kernel time–frequency analysis. Renewable Energy, 66, 468-477. doi: 10.1109/AERO.2004.1368181

Hochmann, D., & Sadok, M. Theory of synchronous averaging Ω. (2004) *Proceedings of IEEE Aerospace Conference, March 6-13,*

J, Kim, (2001). *A Study on the mesh stiffness of helical gear pair and dynamic characteristics of planetary gear train*. Doctoral dissertation. Seoul National University, Korea,http://library.snu.ac.kr/site/snu/viewer/SNUPDF Viewer.jsp?cid=959781&moi=65903&file=1450679

Lei, Y., Kong, D., Lin, J., & Zuo, M. J. (2012). Fault detection of planetary gearboxes using new diagnostic parameters. *Measurement Science and Technology*, 23(5), 055605.

Lebold, M., McClintic, K., Campbell, R., Byington, C., & Maynard, K. (2000). Review of vibration analysis methods for gearbox diagnostics and prognostics. *Proceedings of the 54th Meeting of the Society for Machinery Failure Prevention Technology*. May 1-4, Virginia Beach, VA.

Parker, R. G., & Lin, J. (2003). Mesh phasing relationships in planetary and epicyclic gears. *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 525-534. American Society of Mechanical Engineers. September 2–6

Qu, Y., He, D., Yoon, J., Van Hecke, B., Bechhoefer, E., & Zhu, J. (2014). Gearbox tooth cut fault diagnostics using acoustic emission and vibration sensors—A comparative study. *Sensors*, 14(1), 1372-1393.

Remond, D., & Mahfoudh, J. (2005). From transmission error measurements to angular sampling in rotating machines with discrete geometry. *Shock and Vibration*, 12(2), 149-161.

Samanta, B. (2004). Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mechanical Systems and Signal Processing*, 18(3), 625-644.

Saravanan, N., Cholairajan, S., & Ramachandran, K. I. (2009). Vibration-based fault diagnosis of spur bevel

gear box using fuzzy technique. *Expert Systems with Applications*, 36(2), 3119-3135.

Sirichai, S., Howard, I., Morgan, L., & Teh, K. (1997). Finite element analysis of gears in mesh. *Fifth International Congress on Sound and Vibration*, Australia, pp. 869-876.

Zheng, H., Li, Z., & Chen, X. (2002). Gear fault diagnosis based on continuous wavelet transform. *Mechanical Systems and Signal Processing*, 16(2), 447-457.

Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use

Brian Bole[1], Chetan S. Kulkarni[1], Matthew Daigle[2]

[1] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035*
brian.bole@nasa.gov, chetan.kulkarni@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA 94035*
matthew.j.daigle@nasa.gov

ABSTRACT

Tracking the variation in battery dynamics as a function of health is presently attracting attention in academia and industry due to the increased usage of expensive batteries in dynamic systems such as aircraft and electric cars. The online adaptation of battery models to account for age-dependent changes in dynamics is necessary to maintain accurate estimates of the remaining system operations that can be supported under battery power. A novel method for the adaptation of parameters in an electrochemical model of a lithium-ion battery is presented here. An unscented Kalman filtering algorithm is shown to enable the production of internal battery state estimates and age-dependent electrochemical model parameter estimates using only battery current and voltage data collected over randomized discharge profiles. The use of only data collected over randomized discharge profiles distinguishes this work from other works that make use of reference discharge cycles to judge battery health. The experimental results presented here compare online model estimates produced by the proposed algorithm to offline model estimates obtained by periodically taking batteries offline to run reference discharge cycles.

1. INTRODUCTION

Continued improvements in battery cost, efficiency, and power density have resulted in their increasing use in critical applications such as aircraft and electric cars. In such applications, it is necessary to maintain an accurate model of battery capabilities over many years of use. With an accurate model, precise predictions of end-of-discharge predictions can be made along with predictions of the remaining system operational time that can be supported under battery power (Daigle & Kulkarni, 2013; Saha, Goebel, Poll, & Christophersen, 2009). However, batteries age with increased use, and in order to continue to make accurate predictions, approaches to track of age-dependent changes in battery dynamics are necessary (Saha et al., 2009). While some research has been performed to understand the dynamics of battery aging (Ning & Popov, 2004; Ning, White, & Popov, 2006), relatively little work has been performed to develop approaches for tracking battery age online (Saha & Goebel, 2009).

Modeling methodologies used to represent battery dynamics are generally classified as follows: (*i*) empirical models; (*ii*) electrochemical engineering models; (*iii*) multi-physics models; and (*iv*) molecular/atomist models (Ramadesigan et al., 2012). Empirical models are based on fitting certain functions to past experimental data, without making use of any physicochemical principles. Electrochemical, multi-physics, and atomist models incorporate progressively more fine-grained representations of battery physics. Because more fine-grained models generally increase the model development cost and the cost of computation, it is desired to select a model granularity appropriate to an application's accuracy requirements and available resources (Daigle et al., 2011). In this paper, we use an electrochemistry-based lithium ion (Li-ion) battery model developed in (Daigle & Kulkarni, 2013). The electrochemical modeling used is at level of abstraction high enough that the model is still efficient while improving upon the fidelity of previous approaches (Saha & Goebel, 2009; Daigle et al., 2012; Oliva et al., 2013), which used empirical and equivalent circuit battery models.

The use of unscented Kalman filtering (UKF) (Julier & Uhlmann, 2004) to make online corrections to battery state estimates based on online battery voltage measurements has been described in several recent publications (Daigle & Kulkarni, 2014; Bole et al., 2013; Oliva et al., 2013). The addition of a filtering routine for closed-loop state estimation mitigates the accumulation of model error over time as is seen in open-loop state estimation methods such as the commonly

Brian Bole et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

used method of coulomb counting (Dai et al., 2006). This paper demonstrates the use of UKF not only to estimate the states in an electrochemistry model that vary over a charge-discharge cycle, but also to adapt certain parameters in the model that are known to change as a function of battery age.

While some research has been performed to understand the dynamics of battery aging (Ning & Popov, 2004; Ning et al., 2006), relatively little work has been performed to develop approaches for tracking battery age online (Saha & Goebel, 2009). Generally, a progressive reduction in charge storage capacity and an increase in internal resistance are both know to occur as the battery ages. These changes are typically estimated by compairing the voltage dynamics of healthy and aged batteries over a reference current profile (Broussely et al., 2005). Estimating the state of age-dependent battery parameters from the current-voltage dynamics of batteries in operation is a more challenging proposition than estimating parameters using reference cycles, because individual runs are less able to be directly compaired. This paper introduces experimental results for an algorithm that uses only randomized discharging data to track battery states and estimate model parameters. The experimental results presented here compare online model estimates produced by the proposed algorithm to offline model estimates obtained by periodically taking batteries offline to run reference discharge cycles.

This paper is organized as follows. The electrochemistry-based lithium ion battery model is summarized in Section 2. Battery deterioration modes are discussed in Section 3. Sample results from a set of experiments that age batteries using randomized discharge profiles are introduced in Section 4. A UKF algorithm for online state estimation and age-dependent parameter identification over randomized battery usage periods is described in Section 5. Results generated by applying the UKF algorithm to randomized discharging data sets are summarized in Section 6. Finally, concluding remarks are given in Section 7.

2. BATTERY CHARGE AND DISCHARGE MODELING

A battery converts chemical energy into electrical energy, and often consists of many cells. A cell consists of a positive electrode and a negative electrode with electrolyte in which the ions can migrate. For Li-ion, a common chemistry is a positive electrode consisting of lithium cobalt oxide ($\mathrm{Li}_x\mathrm{CoO}_2$) and negative electrode of lithiated carbon ($\mathrm{Li}_x\mathrm{C}$). These active materials are bonded to metal-foil current collectors at both ends of the cell and electrically isolated by a microporous polymer separator film that is permeable to Li ions. The electrolyte enables lithium ions (Li^+) to diffuse between the positive and negative electrodes. The lithium ions insert or deinsert from the active material depending upon the electrode and whether the active process is charging or discharging, respectively.

This section introduces a battery model derived from a simplified set of electrochemical equations governing charge flow and voltage drops at the cathode, anode, and separator layers of a Li-ion battery. This model is described in detail in (Daigle & Kulkarni, 2013) and summarized here.

The voltage terms of the battery are expressed as functions of the amount of charge in the electrodes (the states of the model). Each electrode, positive (subscript p) and negative (subscript n), is split into two volumes, a surface layer (subscript s) and a bulk layer (subscript b). The differential equations for the battery describe how charge moves through these volumes. The charge (q) variables are described using

$$\dot{q}_{s,p} = i_{app} + \dot{q}_{bs,p} \tag{1}$$
$$\dot{q}_{b,p} = -\dot{q}_{bs,p} + i_{app} - i_{app} \tag{2}$$
$$\dot{q}_{b,n} = -\dot{q}_{bs,n} + i_{app} - i_{app} \tag{3}$$
$$\dot{q}_{s,n} = -i_{app} + \dot{q}_{bs,n}, \tag{4}$$

where i_{app} is the applied electric current The term $\dot{q}_{bs,i}$ describes diffusion from the bulk to surface layer for electrode i, where $i = n$ or $i = p$.

$$\dot{q}_{bs,i} = \frac{1}{D}(c_{b,i} - c_{s,i}), \tag{5}$$

where D is the diffusion constant. The c terms are lithium ion concentrations:

$$c_{b,i} = \frac{q_{b,i}}{v_{b,i}} \tag{6}$$
$$c_{s,i} = \frac{q_{s,i}}{v_{s,i}}, \tag{7}$$

Here, $c_{v,i}$ is the concentration of charge in electrode i, and $v_{v,i}$ is the total volume of charge storage capability. We define $v_i = v_{b,i} + v_{s,i}$. Note now that the following relations hold:

$$q_p = q_{s,p} + q_{b,p} \tag{8}$$
$$q_n = q_{s,n} + q_{b,n} \tag{9}$$
$$q^{\max} = q_{s,p} + q_{b,p} + q_{s,n} + q_{b,n}. \tag{10}$$

We can also express mole fractions (x) based on the q variables:

$$x_i = \frac{q_i}{q^{\max}}, \tag{11}$$
$$x_{s,i} = \frac{q_{s,i}}{q^{\max}_{s,i}}, \tag{12}$$
$$x_{b,i} = \frac{q_{b,i}}{q^{\max}_{b,i}}, \tag{13}$$

where $q^{\max} = q_p + q_n$ refers to the total amount of available Li-ions. It follows that $x_p + x_n = 1$. For Li-ion batteries, when fully charged, $x_p = 0.4$ and $x_n = 0.6$. When fully discharged, $x_p = 1$ and $x_n = 0$ (Karthikeyan, Sikha, & White, 2008).

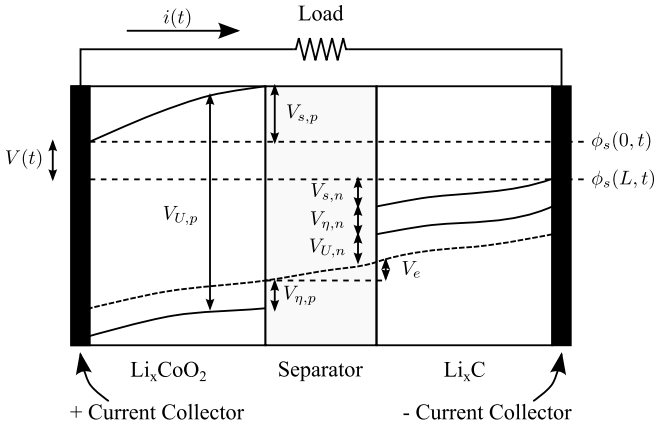


Figure 1. Battery voltages.

The different potentials are summarized in Fig. 1 (originally presented in (Daigle & Kulkarni, 2013) and adapted from (Rahn & Wang, 2013)). The overall battery voltage $V(t)$ is the difference between the potential at the positive current collector, $\phi_s(0,t)$, and the negative current collector, $\phi_s(L,t)$, minus resistance losses at the current collectors (not shown in the diagram). At the positive current collector is the equilibrium potential $V_{U,p}$. This voltage is then reduced by $V_{s,p}$, due to the solid-phase ohmic resistance, and $V_{\eta,p}$, the surface overpotential. The electrolyte ohmic resistance then causes another drop V_e. At the negative electrode, there is a drop $V_{\eta,n}$ due to the surface overpotential, and a drop $V_{s,n}$ due to the solid-phase resistance. The voltage drops again due to the equilibrium potential at the negative current collector $V_{U,n}$. These voltages are described by the following set of equations:

$$V_{U,i} = U_0 + \frac{RT}{nF}\ln\left(\frac{1-x_{s,i}}{x_{s,i}}\right) + V_{\text{INT},i}, \tag{14}$$

$$V_{\text{INT},i} = \frac{1}{nF}\left(\sum_{k=0}^{N_i} A_{i,k}\left((2x_i-1)^{k+1} - \frac{2x_i k(1-x_i)}{(2x_i-1)^{1-k}}\right)\right), \tag{15}$$

$$V_o = i_{app} R_o, \tag{16}$$

$$V_{\eta,i} = \frac{RT}{F\alpha}\text{arcsinh}\left(\frac{J_i}{2J_{i0}}\right), \tag{17}$$

$$J_i = \frac{i}{S_i}, \tag{18}$$

$$J_{i0} = k_i(1-x_{s,i})^{\alpha}(x_{s,i})^{1-\alpha}, \tag{19}$$

$$V = V_{U,p} - V_{U,n} - V_o' - V_{\eta,p}' - V_{\eta,n}', \tag{20}$$

$$\dot{V}_o' = (V_o - V_o')/\tau_o \tag{21}$$

$$\dot{V}_{\eta,p}' = (V_{\eta,p} - V_{\eta,p}')/\tau_{\eta,p} \tag{22}$$

$$\dot{V}_{\eta,n}' = (V_{\eta,n} - V_{\eta,n}')/\tau_{\eta,n}. \tag{23}$$

Here, U_0 is a reference potential, R is the universal gas constant, T is the electrode temperature (in K), n is the number of electrons transferred in the reaction ($n = 1$ for Li-ion), F is Faraday's constant, J_i is the current density, and J_{i0} is the exchange current density, k_i is a lumped parameter of several constants including a rate coefficient, electrolyte concentration, and maximum ion concentration. $V_{\text{INT},i}$ is the activity correction term (0 in the ideal condition). We use the Redlich-Kister expansion with $N_p = 12$ and $N_n = 0$ (see (Daigle & Kulkarni, 2013)). The τ parameters are empirical time constants (used since the voltages do not change instantaneously).

This model contains as states $q_{s,p}$, $q_{b,p}$, $q_{b,n}$, $q_{s,n}$, V_o', $V_{\eta,p}'$, and $V_{\eta,n}'$. The single model output is V. Parameter values for a typical Li-ion cell are given in (Daigle & Kulkarni, 2013).

The state of charge (SOC) of a battery is defined to be 1 when the battery is fully charged and 0 when the battery is fully discharged by convention. In this model, it is analogous to the mole fraction x_n, but scaled from 0 to 1. We distinguish here between nominal SOC and *apparent* SOC (Daigle & Kulkarni, 2013). Nominal SOC is computed based on the combination of the bulk and surface layer control volumes in the negative electrode, whereas apparent SOC is be computed based only on the surface layer. When a battery reaches the voltage cutoff, apparent SOC is 0, and nominal SOC is greater than 0 (how much greater depends on the difference between the diffusion rate and the current drawn). Once the concentration gradient settles out, the surface layer will be partially replenished and apparent SOC will rise while nominal SOC remains the same. Nominal (SOC_n) and apparent (SOC_a) SOC are defined using

$$SOC_n = \frac{q_n}{0.6q^{\max}} \tag{24}$$

$$SOC_a = \frac{q_{s,n}}{0.6q^{\max_{s,n}}}, \tag{25}$$

where $q^{\max_{s,n}} = q^{\max}\frac{v_{s,n}}{v_n}$.[1]

3. BATTERY DETERIORATION MODELING

The rate of deterioration of a battery depends on the chemistry, charge-discharge cycling, temperature, and storage conditions, among other factors. Some relevant physical aging mechanisms observed in batteries are:

1. Solid-electrolyte interface (SEI) layer growth: The negative electrode degrades with the growth of the SEI layer leading to an increase in the impedance. The layers are formed during cycling and storage at high temperatures and entrains the lithium.
2. Lithium corrosion: Lithium in the active carbon material of the negative electrode corrodes over time leading to

[1] Note that SOC of 1 corresponds to the point where $q_n = 0.6q^{\max_{s,n}}$, since the mole fraction at the positive electrode cannot go below 0.4, as described earlier.

degradation. This causes a decrease in the capacity due to irreversible loss of mobile lithium ions.

3. Lithium plating: At low temperatures, high charge rates and low cell voltages form a plating layer on the negative electrode that leads to irreversible loss of lithium.
4. Contact loss: The SEI layer disconnects from the negative electrode, which leads to contact loss and an increase in impedance.
5. Diffusion Stress: Changes in diffusion properties may lead to changes in the charge and discharge times, apparent capacity and impedance.

The various battery aging modes manifest in two major changes to battery electrochemical dynamics. The first is a loss of capacity due to parasitic and side reactions that result in a loss of active (mobile) Li ions. The second is an increase in internal resistance due to SEI layer growth and other factors. Other, less significant, changes to battery electrochemical dynamics are not considered here because the added computational costs are considered to outweigh the benefit to model accuracy. (Ning et al., 2006) looked into loss of active lithium and increase in resistance under constant loading conditions. In this work we look at degradation observed under random loading conditions.

In the battery model, the total available charge in the battery is represented through $q^{\max}$. Therefore, the loss of active material can be represented in the model through a change in $q^{\max}$ (Daigle & Kulkarni, 2013). The R_o parameter captures a constant ohmic drop that does not vary as a function of battery charge.

Figure 2 shows plots of model fitting with a new and aged battery after adding adjustments to the $q^{\max}$ and R_o terms. The figures clearly show the need to tune these parameters to capture the modified electrochemical dynamics of a degraded battery. However, it should also be noted that the fit shown in Figure 2(d) could be improved to a lesser extent by adapting additional terms. The authors suggest that readers interested in adapting additional terms in the electrochemical model start by considering the diffusion rate between the bulk layer and surface layer (D in Eq. (5)). See (Park, Zhang, Chung, Less, & Sastry, 2010) for a discussion of age-related changes to the diffusion rate.

4. A BATTERY AGING EXPERIMENT

This section introduces a battery aging experiment. Battery aging is performed here by repeatedly charging battery cells to approximately 100% SOC ($\approx$4.2 V) and then discharging them to 3.2 V using a randomized sequence of current loads ranging from 0.5A to 4A. The sequence is randomized in order to better represent practical battery usage. After every fifty randomized discharging cycles, an offline characteriza-

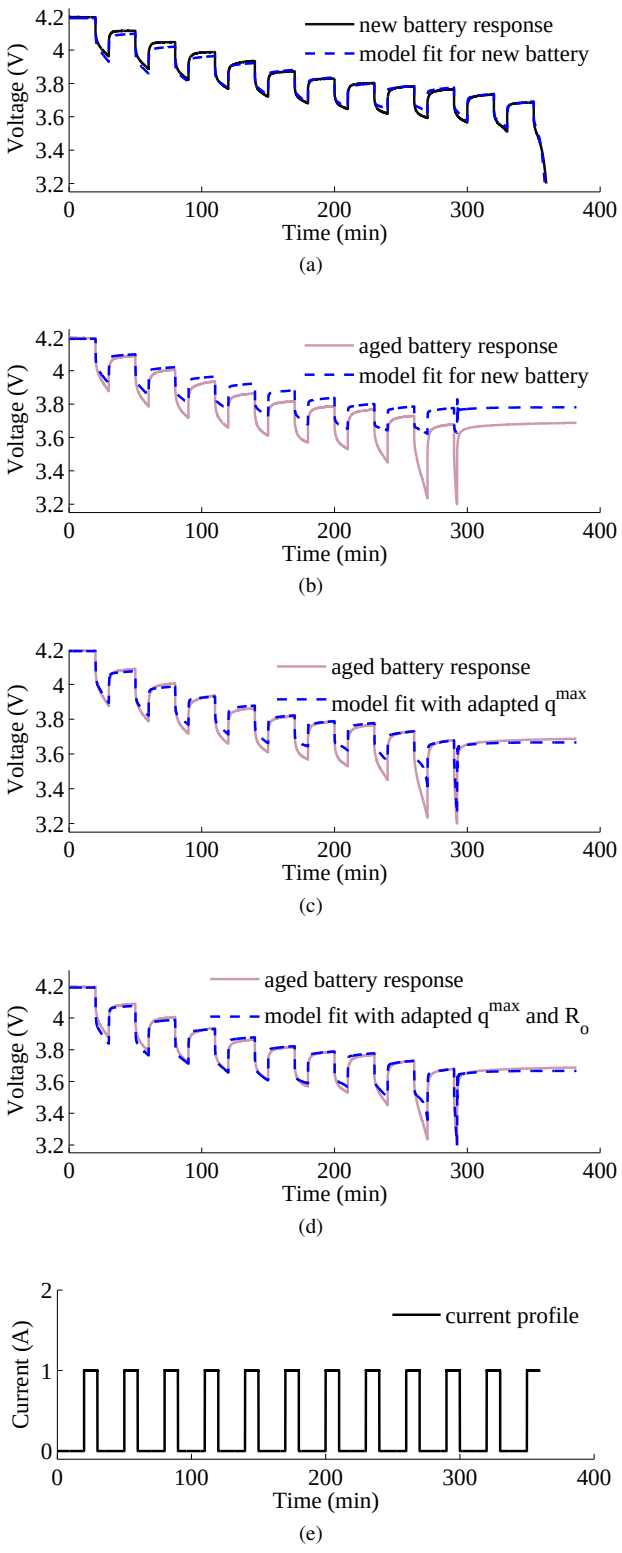


Figure 2. Sample model fitting results for a new battery (a), and an aged battery (b)-(d). The loading profiles used are shown in (e).

Battery Cycling Procedure:

top:

pulsed load characterization:

fully charge to 4.2V
while voltage $>$ 3.2V
rest for 20 min
load at 1A for 10 min
end while
$j = 0$

random walk aging:

while $j < 50$
fully charge to 4.2V
while voltage $>$ 3.2V
$I =$ rand$[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4]$
load at I for 5 min
end while
$j = j + 1$
end while
goto: *top*

Figure 3. Procedure used for battery aging and periodic characterization

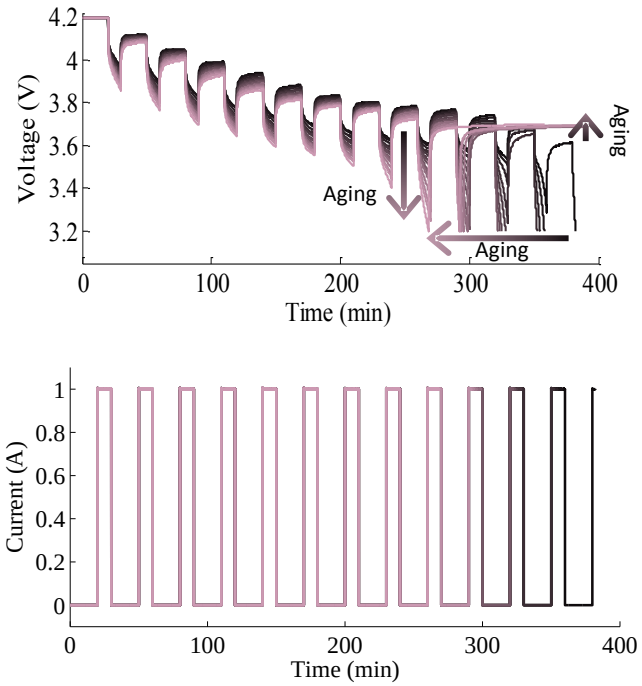


Figure 4. Pulsed voltage profiles recorded periodically over 3 months of continuous battery use.

tion of the q^{max} and R_o model parameters is performed using the pulsed load cycle described in the previous section.

The battery cycling procedure that is used to age individual battery cells and periodically recharacterize health dynamics is outlined in Fig. 3. Fig. 4 shows battery current and voltage for pulsed load characterization cycles taken periodically over about 6 months of continuous battery cycling. Later pulsed load cycles are plotted with lighter line shading.

Age-dependent changes in battery dynamics are denoted with arrows in the figure. The battery voltage is seen to reach the 3.2V cutoff earlier as the battery ages. Aged batteries are also seen to settle to a higher resting voltage after the pulsed profile completes. Both phenomenon can be explained by a decreasing trend in battery capacity and an increasing trend in internal resistance.

Battery capacity loss will result in a decrease in available Li-ions, and therefore a faster discharge time for a given output current, which causes a lowering of surface and bulk battery potentials, see Eqs. (11)-(23). An increase in internal resistance will cause a proportional decrease in battery voltage, see Eq. (16). An increased voltage drop due to an increase in battery internal resistance will also cause the battery voltage to reach the voltage cut-off threshold at a higher SOC, resulting in the higher resting battery voltage measurements seen in Fig. 4.

Fig. 5 shows estimates of q^{max} and R_o obtained by performing an offline least squares fit of the actual and modeled battery voltage over periodic pulsed load characterization cycles. The fitted parameter values are plotted against the integral of battery discharge current, in order to observe the relationship between battery usage and parameter change.

Table 1. Statistics of linear regression fit for q^{max} and R_o

	m_0	m_1	σ^2	R^2
q^{max}	-8.11×10^{-4}	2.15	9.33×10^{5}	0.96
R_o	1.25×10^{-4}	1.05×10^{-1}	1.4×10^{-3}	0.94

A first-order regression model is considered here as a rough approximation of parameter dependence on use. Table 1 shows the slope (denoted m_0), y-intercept (denoted m_1), variance (denoted σ^2), and coefficient of determination (denoted R^2), for the fitted q^{max} and R_o parameters. The coefficient of determination is a normalized measure $\in [0, 1]$ that indicates how well the regression fits the data. A coefficient of determination greater than 0.9 indicates a fairly good model fit. The R^2 values for q^{max} and R_o linear regressions are both seen to exceed this benchmark.

A discussion of battery deterioration modeling and end of useful life prediction using such a model is beyond the scope of this paper. The reader is also cautioned that the battery deterioration observed here is expected to be strongly dependent on the design of experiments.

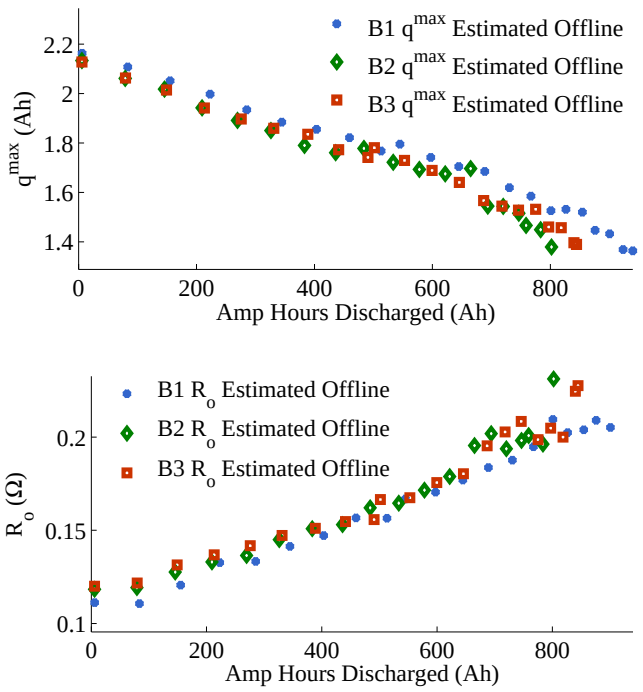


Figure 5. Parameter fitting results for q^{max} and R_o captured periodically over three months of continuous use.

5. ONLINE STATE ESTIMATION AND PARAMETER IDENTIFICATION

An unscented Kalman filter (UKF) (Julier & Uhlmann, 1997, 2004) is introduced here to make corrective updates to the internal state estimates in the battery model in addition to the age-dependent q^{max} and R_o parameters. Among nonlinear filters, the UKF generally has better accuracy than the extended Kalman filter, and avoids the high computational cost of particle filters (Arulampalam, Maskell, Gordon, & Clapp, 2002). We summarize the filter basics here; more details may be found in (Julier & Uhlmann, 1997, 2004).

The UKF assumes the general nonlinear form of the state and output equations, but is restricted to additive Gaussian noise. First, n_s sigma points $\hat{\boldsymbol{\mathcal{X}}}_{k-1|k-1}$ are derived from the current mean $\hat{\mathbf{x}}_{k-1|k-1}$ and covariance estimates $\mathbf{P}_{k-1|k-1}$. The prediction step is:

$$\hat{\boldsymbol{\mathcal{X}}}^i_{k|k-1} = \mathbf{f}(\hat{\boldsymbol{\mathcal{X}}}^i_{k-1|k-1}, \mathbf{u}_{k-1}), i = 1, \ldots, n_s \quad (26)$$

$$\hat{\boldsymbol{\mathcal{Y}}}^i_{k|k-1} = \mathbf{h}(\hat{\boldsymbol{\mathcal{X}}}^i_{k|k-1}), i = 1, \ldots, n_s \quad (27)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_i^{n_s} w^i \boldsymbol{\mathcal{X}}^i_{k|k-1} \quad (28)$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_i^{n_s} w^i \boldsymbol{\mathcal{Y}}^i_{k|k-1} \quad (29)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q} + \sum_i^{n_s} w^i (\boldsymbol{\mathcal{X}}^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\mathcal{X}}^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1})^T, \quad (30)$$

where $\mathbf{Q}$ is the process noise covariance matrix.

The update step is:

$$\mathbf{P}_{yy} = \mathbf{R} + \sum_i^{n_s} w^i (\boldsymbol{\mathcal{Y}}^i_{k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\boldsymbol{\mathcal{Y}}^i_{k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T \quad (31)$$

$$\mathbf{P}_{xy} = \sum_i^{n_s} w^i (\boldsymbol{\mathcal{X}}^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\mathcal{Y}}^i_{k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T \quad (32)$$

$$\mathbf{K}_k = \mathbf{P}_{xy}\mathbf{P}^{-1}_{yy} \quad (33)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \quad (34)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{yy}\mathbf{K}^T_k, \quad (35)$$

where $\mathbf{R}$ is the sensor noise covariance matrix.

The use of the UKF for closed-loop state updates of the 7 states in the battery model described in Section 2, was presented in (Daigle & Kulkarni, 2013). The UKF algorithm presented in (Daigle & Kulkarni, 2013) was updated for use here by considering the R_o parameter in Eq. 16 as an additional state to be updated online by the UKF.

An additional outer-loop process is then used to infer q^{max} values that correspond to a given window of SOC_n estimates under known battery loading conditions. We elected to use an outer-loop estimation process for q^{max}, rather than including it in the UKF because it is straightforward to infer q^{max} from SOC_n estimates. This is seen by first rewriting the SOC_n definition, given in Eq. (24), in terms of a UKF-based estimate of q_n.

$$\widehat{SOC_n}(t) = \frac{\widehat{q_n}(t)}{0.6q^{max}}, \quad (36)$$

where $\widehat{q_n}(t)$ represents an estimate of q_n at time t, and $\widehat{SOC_n}(t)$ represents a subsequently derived estimate of SOC_n. The difference in $\widehat{SOC_n}$ estimates of over a given time window is then expressed as:

$$\widehat{SOC_n}\Big|_{t_0}^{t} = \frac{\widehat{q_n}|_{t_0}^{t}}{0.6q^{max}}. \quad (37)$$

Next, consider that the true value of $q_n|_{t_0}^{t}$ is equal to the amount of charge flow into or out of the battery over the given time window.

$$q_n|_{t_0}^{t} = \int_{t_0}^{t} i_{app}, \quad (38)$$

where i_{app} represents the net current into or out of the battery.

A substitution of Eq. (38) into Eq. (37) yeilds an inferred estimate of q^{max}.

$$\hat{q}^{max}(t) = \frac{q_n|_{t_0}^{t}}{0.6\ \widehat{SOC_n}\Big|_{t_0}^{t}}, \tag{39}$$

where $\hat{q}^{max}(t)$ represents an estimate of the q^{max} model parameter at time t.

6. RESULTS

Fig. 6(a) shows an example of the online adaptation of battery state estimates and model parameters in order to match the measured voltage response of an aged battery over a randomized discharge cycle. The predicted voltage response for a new battery, and the voltage estimation output of a UKF-based observer initialized with the parameters of a new battery are also plotted in Fig. 6(a). Online UKF estimates of the q^{max} and R_o parameters are shown in Figs. 6(b) and (c). The randomized loading profile used in this example is shown in Fig. 6(d).

The battery voltage output estimates from the UKF are seen to converge to match the measured voltage estimates over the 40 minute randomized discharging cycle. The variation seen in the q^{max} and R_o estimates from 0 to 40 minutes is due primarily to the large initial disparity between the parameters fitted for a new battery model and the model parameters needed to explain the dynamics of an aged battery. Typically, the model parameters estimated by the UKF over a previous discharge cycle would be used to initialize the battery model for the following discharge cycle. This would lead to a much smaller error in the initial parameter estimates and less parameter variation would result.

Fig. 7 shows online q^{max} and R_o estimates produced by the UKF observer over successive randomized discharge cycles. The offline estimates of q^{max} and R_o that were originally shown in Fig. 5 are also plotted in Fig. 7 for comparison. The online q^{max} estimates are seen in Fig. 7 to track the offline q^{max} estimates very closely. This indicates not only that the UKF is able to track battery capacity over randomized discharging cycles, but also that the online battery SOC estimates that are used to calculate capacity (see Eqs. (36)-(39)) are also tracking the true battery SOC over randomized usage.

Online R_o estimates are seen in Fig. 7 to be noticeably lower and more non-linear than the offline R_o estimates. Despite this discrepancy, online R_o estimates display some similarities to the offline estimates. Both sets of R_o estimates tend be monotonically increasing with battery age, and both show a slightly lower resistance estimate for battery $B1$ than for $B2$ and $B3$. The bias observed between offline and online estimates can be attributed primarily to a difficulty in setting up the process noise covariance matrix, $\mathbf{Q}$, in the UKF to filter

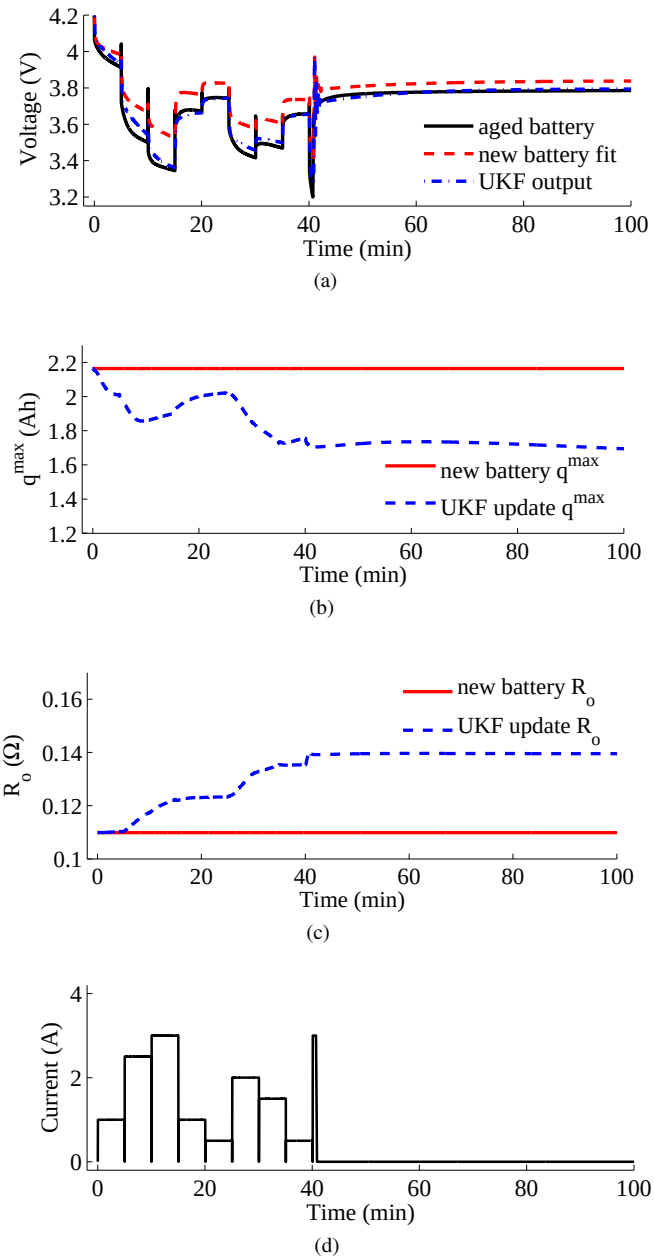


Figure 6. The measured voltage response of an aged battery over a randomized discharge cycle, the predicted voltage response for a new battery, and the voltage estimation output of a UKF-based observer are shown in (a). Online estimates of the q^{max} and R_o parameters are shown in (b) and (c). The loading profile used is shown in (d).

out the effects of the R_o term from those of the other parameters in state vector. It is certain that a refinement of $\mathbf{Q}$ could improve the tracking performances observed for the q^{max} and R_o parameters. However, the non-optimized tracking performance shown here is sufficient to demonstrate the feasibility of the proposed approach for model adaptation over variable battery usage.

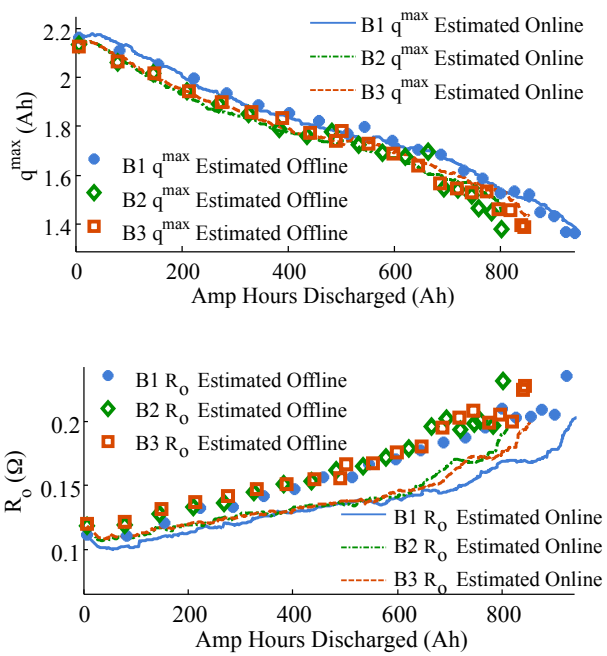


Figure 7. Online and offline estimates of q^{max} and R_o model parameters.

7. CONCLUSIONS

An approach for the online tracking of age-dependent changes in battery dynamics was presented. An electrochemistry-based Li-ion battery model was shown to relate known age-dependent electrochemical phenomena to changes in battery input-output dynamics observed over randomized battery usage. A battery aging experiment was introduced, and an unscented Kalman filtering algorithm was shown to track age-dependent changes in battery model parameters over successive randomized battery discharging profiles.

In future work the battery sate of health tracking approach presented here may be extended to the online prediction of remaining useful life. This would require additional modeling of the underlying physics of battery degradation as a function of usage. Linear regression models for battery capacity and internal resistance change as a function of energy discharged are analyzed here as a starting point for this future work.

ACKNOWLEDGMENT

The project support by NASA's AvSafe/SSAT and OCT/ACLO are respectfully acknowledged.

REFERENCES

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Bole, B., Teubert, C., Chi, Q. C., Edward, H., Vazquez, S., Goebel, K., & Vachtsevanos, G. (2013). SIL/HIL replication of electric aircraft powertrain dynamics and inner-loop control for V&V of system health management routines. In *Annual conference of the prognostics and health management society.*

Broussely, M., Biensan, P., Bonhommeb, F., Blanchard, P., Herreyre, S., Nechev, K., & Staniewicz, R. (2005). Main aging mechanisms in li ion batteries. *Journal of Power Sources*, *146*(1-2), 90-96.

Dai, H., Wei, X., & Sun, Z. (2006). Online soc estimation of high-power Lithium-Ion batteries used on HEVs. In *IEEE international conference on vehicular electronics and safety.*

Daigle, M., & Kulkarni, C. (2013, October). Electrochemistry-based battery modeling for prognostics. In *Annual conference of the prognostics and health management society 2013* (p. 249-261).

Daigle, M., & Kulkarni, C. (2014). A battery health monitoring framework for planetary rovers. In *Proceedings of the IEEE aerospace conference.*

Daigle, M., Roychoudhury, I., Narasimhan, S., Saha, S., Saha, B., & Goebel, K. (2011, September). Investigating the effect of damage progression model choice on prognostics performance. In *Proceedings of the annual conference of the prognostics and health management society 2011* (p. 323-333).

Daigle, M., Saxena, A., & Goebel, K. (2012, September). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society* (p. 326-335).

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th international symposium on aerospace/defense sensing, simulation and controls* (pp. 182–193).

Julier, S. J., & Uhlmann, J. K. (2004, Mar). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Karthikeyan, D. K., Sikha, G., & White, R. E. (2008). Thermodynamic model development for lithium intercalation electrodes. *Journal of Power Sources*, *185*(2), 1398–1407.

Ning, G., & Popov, B. N. (2004). Cycle life modeling of lithium-ion batteries. *Journal of The Electrochemical Society*, *151*(10), A1584–A1591.

Ning, G., White, R. E., & Popov, B. N. (2006). A generalized cycle life model of rechargeable li-ion batteries. *Electrochimica Acta*, *51*(10), 2012–2022.

Oliva, J., Weihrauch, C., & Bertram, T. (2013, October). A model-based approach for predicting the remaining

driving range in electric vehicles. In *Annual conference of the prognostics and health management society 2013* (p. 438-448).

Park, M., Zhang, X., Chung, M., Less, G. B., & Sastry, A. M. (2010). A review of conduction phenomena in li-ion batteries. *Journal of Power Sources*, *195*(24), 7904–7929.

Rahn, C. D., & Wang, C.-Y. (2013). *Battery systems engineering*. Wiley.

Ramadesigan, V., Northrop, P. W., De, S., Santhanagopalan, S., Braatz, R. D., & Subramanian, V. (2012). Modeling and simulation of lithium-ion batteries from a systems engineering perspective. *Journal of The Electrochemical Society*, *159*(3), R31–R45.

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society 2009.*

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009, February). Prognostics methods for battery health monitoring using a Bayesianframework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291–296.

BIOGRAPHIES

Brian M. Bole graduated from the FSU-FAMU School of Engineering with a B.S. in Electrical and Computer Engineering and a B.S. in Applied Math. Brian received M.S. and Ph.D. degrees in Electrical Engineering from the Georgia Institute of Technology. His research interests include: analysis of stochastic processes, risk analysis, and optimization of stochastic systems. Brian is currently investigating the use of risk management and stochastic optimization techniques for prognostics and prognostics-informed decision making in robotic and aviation applications. From 2011 to 2013 he performed joint research with the Prognostic Center of Excellence at NASA Ames under the NASA graduate student research fellowship. He is currently working as a research engineer for Stinger Ghaffarian Technologies and is conducting joint research with the intelligent systems division at NASA Ames.

Chetan S. Kulkarni received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. He was a Senior Project Engineer with Honeywell Automation India Limited (HAIL) from 2003 till April 2006. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he has been a Staff Researcher with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and the IEEE.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

Application of Unscented Kalman Filter for Condition Monitoring of an Organic Rankine Cycle Turbogenerator

Leonardo Pierobon[1], Rune Schlanbusch[2], Rambabu Kandepu[3] and Fredrik Haglind[4]

[1,4] *Department of Mechanical Engineering, Technical University of Denmark*
Building 403, 2800 Kongens Lyngby, Denmark
lpier@mek.dtu.dk
frh@mek.dtu.dk

[2,3] *Teknova AS*
Gimlemoen 19, 4630 Kristiansand, Norway
Rune.Schlanbusch@teknova.no
Rambabu.Kandepu@teknova.no

ABSTRACT

This work relates to a project focusing on energy optimization on offshore facilities. On oil and gas platforms it is common practice to employ gas turbines for power production. So as to increase the system performance and reduce emissions, a bottoming cycle unit can be designed with particular emphasis on compactness and reliability. In such context, organic Rankine cycle turbogenerators are a promising technology. The implementation of an organic Rankine cycle unit is thus considered for the power system of the Draugen offshore platform in the northern sea, which is the case study for this project. Considering the plant dynamics, it is of paramount importance to monitor the peak temperatures within the once-through boiler serving the bottoming unit to prevent the decomposition of the working fluid. This paper accordingly aims at applying the unscented Kalman filter to estimate the temperature distribution inside the primary heat exchanger by engaging a detailed and distributed model of the system and available measurements. Simulation results prove the robustness of the unscented Kalman filter with respect to process noise, measurement disturbances and initial conditions.

1. INTRODUCTION

Owing to environmental concerns and with increasing incentives for reducing CO_2 emissions and pollutants offshore, optimization of energy usage on oil and gas facilities has become a focus area. On offshore platforms one or more redundant gas turbines supply the electric power demand. As an example, a standard operational strategy is to share the load between two engines, while a third is on stand-by or on maintenance. The two gas turbines typically run at fairly low loads (around 50%) in order to decrease the risk of failure of the system, which would cause a high economic loss to the platform operator. On the other hand, this operational strategy reduces significantly the system performance, which in turns results in a large amount of waste heat contained in the exhaust gases exiting the engines (Nguyen et al., 2013).

A viable solution to enhance the efficiency is to implement a waste heat recovery unit at the bottom of the gas turbines. A mature technology accomplishing this duty is the Steam Rankine Cycle (SRC) power module. (Kloster, 1999) described the existing SRC units in the Oseberg, Eldfisk and Snorre B offshore installations. Air Bottoming Cycles (ABCs) constitute a valid alternative to SRC units as they employ a non-toxic and inflammable working fluid. Moreover, ABC power modules do not require a condenser as they operate as open-cycles, thus leading to high compactness and low weight. (Bolland, Forde, & Hånde, 1996) carried out a feasibility study on the implementation of ABCs offshore. Results proved that, despite the low gain in performance, low weight and short pay-back time can be achieved. (Pierobon, Nguyen, Larsen, Haglind, & Elmegaard, 2013) proposed instead the use of Organic Rankine Cycle (ORC) power modules by tailoring their design to the Draugen oil and gas facility. For the same case study, (Pierobon, Haglind, Kandepu, Fermi, & Rossetti, 2013) demonstrated that the use of ORC units provides larger economic revenues and power system performances compared to ABC and SRC modules.

While ORC turbogenerators work in principle similarly to steam Rankine cycle units, the working fluid is instead an

Leonardo Pierobon et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

organic compound characterized by lower critical temperatures and pressures than water, thus making these systems suitable for low and medium temperature waste heat recovery (Quoilin, Broek, Declaye, Dewallef, & Lemort, 2013). As a drawback, organic fluids may experience chemical deterioration and decomposition at high temperatures. This criticality is owed to the breakage of chemical bonds between the molecules and the formation of smaller compounds, which can then react to create other hydrocarbons. As the system performance strongly relates to transport and physical properties of the working fluid, those chemical phenomena can severely reduce the net power output and the components' lifetime. In such context, monitoring the temperature profiles inside the heat exchangers serving the ORC unit is a pivotal aspect to enhance plant reliability and reduce maintenance periods.

A possible solution accomplishing these tasks is the Kalman Filter (KF), an algorithm which employs a state space model of the system and measurements ascertained over time, containing noise and disturbances, and provides estimates of unknown variables that are usually more accurate than those based on a single measurement alone. As examples of applications to heat exchangers, (G. Jonsson & Palsson, 1994) used the KF algorithm to adjust generic empirical correlations commonly employed to estimate the heat transfer coefficients, while (Loparo, Buchner, & Vasudeva, 1991) proposed a non-linear KF algorithm for leak detection in an experimental laboratory heat exchanger process. More recently, (G. R. Jonsson, Lalot, Palsson, & Desmet, 2007) demonstrated the use of an Extended Kalman Filter (EKF) to detect fouling in heat exchangers.

Notwithstanding the aforementioned works, to the knowledge of the authors the KF algorithm has not yet been applied to ORC waste heat recovery systems. This paper accordingly aims at demonstrating the use of the Kalman filter to estimate the temperature distribution in the primary heat exchanger of an ORC unit, which is used to augment the performance of the gas turbine-based power system installed on the Draugen oil and gas platform. It is reported that Unscented Kalman Filter (UKF) performs better in estimating the state variables in a non-linear system in comparison with EKF (Kandepu, Foss, & Imsland, 2008). Accordingly in this article, the UKF is applied to the ORC unit with focus on the primary heat exchanger. A state space model of the ORC system based on first principles is developed. Disturbances are assumed for measurements of temperature, mass flow and density utilizing in-silico simulation-based data and assuming a Gaussian distribution. The UKF is thus applied to estimate the temperature distribution inside the main heat exchanger while the remaining variables are assumed to be measurable.

This paper is structured as follows: Section 2 introduces the case study, and Section 3 describes the state space model of the ORC turbogenerator. The unscented Kalman filter algorithm is then outlined in Section 4, while the results are reported and discussed in Section 5. Concluding remarks are given in Section 6.

2. CASE STUDY

The case study is the power system installed on the Draugen oil and gas offshore platform, located 150 km from Kristiansund, in the Norwegian Sea. The reservoir was discovered in 1984 and started operation in 1993. The platform, operated by A/S Norske Shell, produces gas exported via Åsgard gas pipeline to Kårstø (Norway) and oil, which is first stored in tanks at the bottom of the sea and then exported via a shuttle tanker (once every 1-2 weeks). The normal power demand is around 19 MW and it can increase up to 25 MW during oil export. To enhance the reliability and to diminish the risk of failure of the power system, two turbines run at a time covering 50% of the load each, while the third one is kept on standby, allowing for maintenance work. Despite the low performance, this strategy ensures the necessary reserve power for peak loads, and the safe operation of the engines.

Figure 1 shows the layout of the power system with the additional ORC turbogenerator recovering the heat contained in the exhaust gases produced by gas turbine A. Gas turbines B and C are not reported. Note that the bottoming cycle units should have the capability to harvest the waste heat alternatively from the other two engines, thus ensuring high performances when switching the gas turbines on operation. The twin-spool engine employs two coaxial shafts coupling the Low Pressure Compressor (LPC) with the Low Pressure Turbine (LPT) and the High Pressure Compressor (HPC) with the High Pressure Turbine (HPT). The Power Turbine (PT) transfers mechanical power through a dedicated shaft to the electric Generator (GEN). Natural gas is the fuel utilized in the Combustion Chamber (CC).

The ORC unit comprehends the single-pressure non-reheat Once-Through Boiler (OTB), the turbine, the sea-water cooled shell-and-tube condenser and the feed-water pump. The working fluid is cyclopentane (C_5H_{10}), a compound widely adopted for operating ORC systems in this range of temperature, see e.g. (Del Turco et al., 2011). As the slope of the saturation curve of cyclopentane is positive (dry fluid), a shell-and-tube recuperator is added to decrease the energy contained in the superheated vapour exiting the ORC expander. Figure 2 illustrates the $T - s$ diagrams of the ORC power unit considered in this study. Starting from point 3, cyclopentane is first preheated ($3 \rightarrow 4$), vaporized ($4 \rightarrow 5$) and superheated ($5 \rightarrow 6$) in the once-through boiler. The fluid is then expanded in the turbine ($6 \rightarrow 7$) , and cooled down in the recuperator ($7 \rightarrow 8$). In this manner the inlet temperature in the OTB can be enhanced by recovering energy from the superheated vapour exiting the turbine. The working fluid is

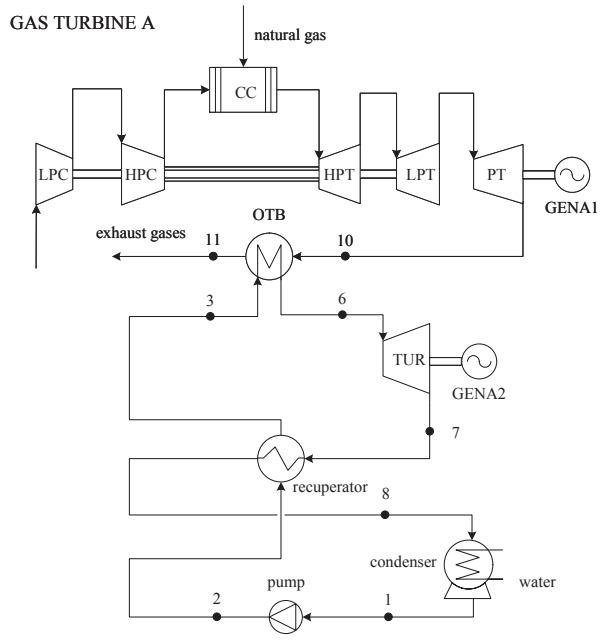


Figure 1. Simplified layout of the power system on the Draugen offshore oil and gas platform. Gas turbine B and C are not shown. The organic Rankine cycle module recuperates part of the thermal power released with the exhaust gases of one engine, in the case gas turbine A.

then condensed ($8 \rightarrow 9 \rightarrow 1$) and pumped up ($1 \rightarrow 2$) to the highest pressure level through the cold side of the recuperator ($2 \rightarrow 3$), thus closing the cycle. Note that Figure 1 does not report node 9 as the saturated vapour condition occurs inside the condenser.

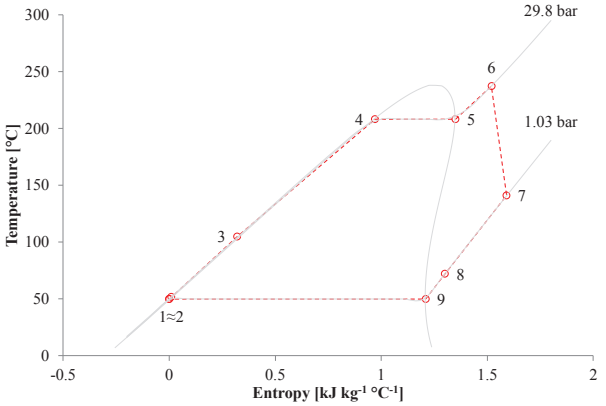


Figure 2. Saturation curve of cyclopentane in a $T-s$ diagram, showing the thermodynamic cycle state points of the organic Rankine cycle system.

3. STATE SPACE MODEL

The transient performance of ORC power systems is primarily driven by the thermal inertia of the heat exchangers. Figure 3 illustrates the discretized model utilized for the once-through boiler and the recuperator. The model features a 1D flow model for the hot side (top) and cold side (bottom), and the 1D thermal model for the tube walls (middle). A counter-flow configuration and uniform pressure distribution are assumed. The tube metal wall is modelled by a 1D dynamic

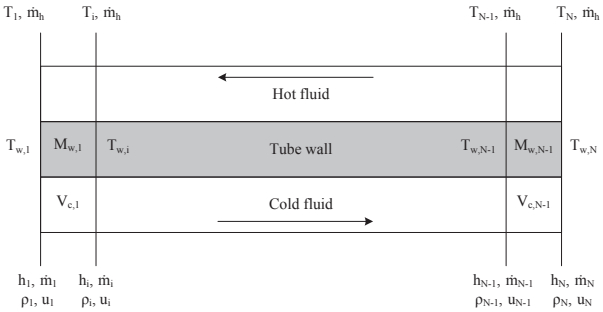


Figure 3. Heat exchanger discretized model.

heat balance equation, which for the i-cell can be written as

$$M_{\mathrm{w,i}} c_{\mathrm{w}} \frac{d\bar{T}_{\mathrm{w,i}}}{dt} = \dot{q}_{\mathrm{h}} - \dot{q}_{\mathrm{c}} \,, \tag{1}$$

where $M_{\mathrm{w,i}}$ and c_{w} are the mass and the heat capacity of the metal wall, and $\bar{T}_{\mathrm{w,i}}$ is the wall temperature at the i-volume, calculated as the arithmetic average between the temperatures at the inner and outer node. The variable $\dot{q}_{\mathrm{h}}$ is the heat provided by the hot stream and $\dot{q}_{\mathrm{c}}$ is the heat transferred to the cold side. The flow model for the cold side contains one-dimensional dynamic mass and energy balance equations, which can be expressed as

$$V_{\mathrm{c,i}} \frac{d(\bar{u}_{\mathrm{i}} \bar{\rho}_{\mathrm{c,i}})}{dt} = \dot{m}_{\mathrm{i}} h_{\mathrm{i}} - \dot{m}_{\mathrm{i+1}} h_{\mathrm{i+1}} + \dot{q}_{\mathrm{c}} \,, \tag{2}$$

$$V_{\mathrm{c,i}} \frac{d\bar{\rho}_{\mathrm{c,i}}}{dt} = \dot{m}_{\mathrm{i}} - \dot{m}_{\mathrm{i+1}} \,, \tag{3}$$

where $\dot{m}_{\mathrm{i}}$ and h_{i} represent the mass flow and the enthalpy at the i-node. The variables $\bar{u}_{\mathrm{c,i}}$ and $\bar{\rho}_{\mathrm{c,i}}$ are the internal specific energy and the density of the volume $V_{\mathrm{c,i}}$ calculated as arithmetic average between the values at the inner and outer node. In light of the relatively small variations with time of the thermodynamic properties on the gas side, steady state mass and energy balances are considered (see Equation 4).

$$\dot{q}_{\mathrm{h}} = c_{\mathrm{p,h}} \dot{m}_{\mathrm{h}} (T_{\mathrm{i+1}} - T_{\mathrm{i}}) \,. \tag{4}$$

Owing to their relatively small contributions, the thermal resistance in the radial direction and thermal diffusion in the axial direction are neglected. For the once-through boiler and the recuperator the heat transfer coefficient between the hot and the outer pipe surface is much lower than the one be-

tween the inner pipe surface and the ORC working fluid flow. Therefore, the overall heat transfer is essentially dependent on the hot side only, and the working fluid temperature is always close to the inner surface temperature of the pipe.

The heat transfer coefficient at the interface between the hot and the metal wall, in off-design conditions, is evaluated with the relation (Incropera, DeWitt, Bergman, & Lavine, 2007)

$$U = U_{\text{des}} \left(\frac{\dot{m}}{\dot{m}_{\text{des}}} \right)^{\gamma} , \tag{5}$$

where U is the heat transfer coefficient and the subscript "des" refers to the value at nominal operating conditions. The exponent γ, taken equal to 0.6, is the exponent of the Reynolds number in the heat transfer correlation. The thermal interaction between the wall and the cold stream is described by specifying a sufficiently high constant heat transfer coefficient, so that the fluid temperature is close to the wall temperature, and the overall result is dominated by the hot side heat transfer.

For the turbine the Stodola's cone law, expressing the relation between the pressure at the inlet and the outlet of the expander with the mass flow rate and the turbine inlet temperature is applied (Stodola, 1922). To predict the turbines off-design performance, the correlation relating the isentropic efficiency and the non-dimensional flow coefficient proposed by (Schobeiri, 2005) is utilized.

The isentropic efficiency of the pump in part-load is derived using the methodology proposed by (Veres, 1994), while the part-load characteristic of the electric generator is modelled using the equation suggested by (Haglind & Elmegaard, 2009).

Table 1 lists the parameters employed to parametrize the state space model of the ORC turbogenerator. The condensing pressure of the working fluid is fixed to 1.03 bar, corresponding to a temperature of 50 °C, so as to avoid inward air leakage into the condenser. The weight, volume and UA-values of the once-through boiler and the recuperator are obtained using an in-house simulation tool (Pierobon, Casati, Casella, Haglind, & Colonna, 2014), which has been extensively validated with public domain data.

Table 1. Design-point variables utilized to parametrize the state space model of the organic Rankine cycle system.

Component	Parameters
Gas turbine	
Exhaust temperature t_{10}	$379.2\,°C$
Exhaust mass flow $\dot{m}_{10}$	$91.5\,kg\,s^{-1}$
Once-through boiler	
Volume (cold side)	$4\,m^3$
Weight (metal walls)	$50\,ton$
UA-value	$400\,kW\,K^{-1}$
Recuperator	
Volume (cold side)	$2\,m^3$
Weight (metal walls)	$5\,ton$
UA-value	$209.6\,kW\,K^{-1}$
Turbine	
Stodola constant	$30.4\,kg\,s^{-1}\,K^{0.5}\,bar^{-1}$
Isentropic efficiency	0.80
Electric generator efficiency	0.98
Pump	
Delivery pressure p_2	$29.8\,bar$
Inlet pressure p_1	$1.03\,bar$
Isentropic efficiency	0.72

4. UNSCENTED KALMAN FILTER ALGORITHM

In this section we will present the algorithm for the UKF for a general non-linear system. Let the system be represented by the following general non-linear discrete time equations

$$x_k = f(x_{k-1}, v_{k-1}, u_{k-1}) , \tag{6}$$

$$y_k = h(x_k, n_k, u_k) , \tag{7}$$

where $x \in \mathbb{R}^{n_x}$ is the system state, $v \in \mathbb{R}^{n_v}$ the process noise, $n \in \mathbb{R}^{n_n}$ the observation noise, u the input and y the noisy observation of the system.

The UKF algorithm is presented in the following (Julier & Uhlmann, 1997), (Wan & Van Der Merwe, 2000). Let the system be represented by (6) and (7). An augmented state at time instant k is defined

$$x_k^a \triangleq \begin{bmatrix} x_k \\ v_k \\ n_k \end{bmatrix} .$$

The augmented state dimension is,

$$N = n_x + n_v + n_n . \tag{8}$$

Similarly, the augmented state covariance matrix is built from the covariance matrices of x, v and n according to

$$P^a \triangleq \begin{bmatrix} P_x & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix} ,$$

where P_v and P_n are the process and observation noise covariance matrices. Note that the augmented state is needed for non-additive noise; for additive noise the original state vector is sufficient.

$Initialization\ at\ k = 0:$

$$\begin{aligned} \widehat{x}_0 &= E[x_0] , \quad P_{x_0} = E\left[(x_0 - \widehat{x}_0)(x_0 - \widehat{x}_0)^T \right] , \\ \widehat{x}_0^a &= E[x^a] = E[\widehat{x}_0 \quad 0 \quad 0]^T , \\ P_0^a &= E\left[(x_0^a - \widehat{x}_0^a)(x_0^a - \widehat{x}_0^a)^T \right] = \begin{bmatrix} P_x & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix} , \end{aligned}$$

For $k = 1, 2, ...\infty$:

Generate Sigma-points
Calculate $2N + 1$ sigma-points based on the present state covariance:

$$\mathbf{X}^a_{i,k-1} \begin{cases} \triangleq \widehat{x}^a_{k-1}, & i = 0 \\ \triangleq \widehat{x}^a_{k-1} + \gamma \mathbf{S}_i, & i = 1, ..., N \\ \triangleq \widehat{x}^a_{k-1} - \gamma \mathbf{S}_i, & i = N+1, ..., 2N \end{cases} , \quad (9)$$

where $\mathbf{S}_i$ is the ith column of the matrix,

$$S = \sqrt{P^a_{k-1}}.$$

In (9) γ is a scaling parameter

$$\gamma = \sqrt{N + \lambda}, \quad \lambda = \alpha^2(N + \kappa) - N \, ,$$

where α and κ are tuning parameters. We must choose $\kappa \geq 0$, to guarantee the semi-positive definiteness of the covariance matrix, a good default choice is $\kappa = 0$. The parameter α, $0 \leq \alpha \leq 1$, controls the size of the sigma-point distribution and it should ideally be a small number.

The ith sigma point (augmented) is the ith column of the sigma point matrix,

$$\mathbf{X}^a_{i,k-1} = \begin{bmatrix} \mathbf{X}^x_{i,k-1} \\ \mathbf{X}^v_{i,k-1} \\ \mathbf{X}^n_{i,k-1} \end{bmatrix} ,$$

where the superscripts x, v and n refer to a partition conformal to the dimensions of the state, process noise and measurement noise respectively.

Time-update equations
Transform the sigma points through the state-update function,

$$\mathbf{X}^x_{i,k/k-1} = f\left(\mathbf{X}^x_{i,k-1}, \mathbf{X}^v_{i,k-1}, u_{k-1}\right), \; i = 0, 1, ..., 2N \, .$$

Calculate the *a priori* state estimate and *a priori* covariance,

$$\begin{aligned} \widehat{x}^-_k &= \sum_{i=0}^{2N} \left(w^{(i)}_m \mathbf{X}^x_{i,k/k-1} \right) , \\ P^-_{x_k} &= \sum_{i=0}^{2N} w^{(i)}_c \left(\mathbf{X}^x_{i,k/k-1} - \widehat{x}^-_k \right) \left(\mathbf{X}^x_{i,k/k-1} - \widehat{x}^-_k \right)^T . \end{aligned}$$

The weights $w^{(i)}_m$ and $w^{(i)}_c$ are defined as,

$$\begin{aligned} w^{(0)}_m &= \frac{\lambda}{N + \lambda}, \;\; i = 0, \\ w^{(0)}_c &= \frac{\lambda}{N + \lambda} + (1 - \alpha^2 + \beta), \;\; i = 0, \\ w^{(i)}_m &= w^{(i)}_c = \frac{1}{2(N + \lambda)}, \;\; i = 1, ..., 2N, \end{aligned}$$

where β is a non-negative weighting parameter introduced to affect the weighting of the $zero$th sigma-point for the calculation of the covariance. This parameter (β) can be used to incorporate knowledge of the higher order moments of the distribution. For a Gaussian prior a typical choice is $\beta = 2$, as suggested by (Wan & Van Der Merwe, 2000).

Measurement-update equations
Transform the sigma points through the measurement-update function,

$$\mathbf{Y}_{i,k/k-1} = h\left(\mathbf{X}^x_{i,k/k-1}, \mathbf{X}^n_{k-1}, u_k\right), \; i = 0, 1, ..., 2N \, ,$$

and the mean and covariance of the measurement vector is calculated,

$$\begin{aligned} \widehat{y}^-_k &= \sum_{i=0}^{2N} w^{(i)}_m \mathbf{Y}_{i,k/k-1} \, , \\ P_{\overline{y}_k} &= \sum_{i=0}^{2N} w^{(i)}_c \left(\mathbf{Y}_{i,k/k-1} - \widehat{y}^-_k \right) \left(\mathbf{Y}_{i,k/k-1} - \widehat{y}^-_k \right)^T . \end{aligned}$$

The cross covariance is calculated according to

$$P_{x_k y_k} = \sum_{i=0}^{2N} w^{(i)}_c \left(\mathbf{X}^x_{i,k/k-1} - \widehat{x}^-_k \right) \left(\mathbf{Y}_{i,k/k-1} - \widehat{y}^-_k \right)^T .$$

The Kalman gain is given by,

$$K_k = P_{x_k y_k} P^{-1}_{\overline{y}_k},$$

and the UKF estimate and its covariance are computed from the standard Kalman update equations,

$$\widehat{x}_k = \widehat{x}^-_k + K_k \left(y_k - \widehat{y}^-_k \right),$$

$$P_{x_k} = P^-_{x_k} - K_k P_{\overline{y}_k} K^T_k .$$

5. SIMULATION RESULTS

In this section we present the simulation results derived by applying the UKF (see Section 4) on the ORC model described in Section 3 and parametrized according to Table 1. A static model of the gas turbine is included in the form of static functions to simulate the exhaust gas temperature and mass flow of the engine, *i.e.* the temperature and mass flow at node 10 in Figure 1. The static model serves as input to the ORC model and it is specified in terms of gas turbine load set-point. The ORC model was implemented on the form (6)–(7). The state of the fluid at each node from 1 to 8 (see Figure 1), is characterized by three state variables; mass flow $\mathbf{x}_{\dot{m}}$, density $\mathbf{x}_d$ and temperature $\mathbf{x}_T$, thus $\mathbf{x} = [\mathbf{x}^\top_T \; \mathbf{x}^\top_{\dot{m}} \; \mathbf{x}^\top_d]^\top$. The once-through boiler was discretized into 10 volumes with input and output at node 3 and 6, respectively (see Figure 2). All the state variables, *i.e.* mass flow, density and temperature of cyclopentane, are assumed to be measurable at all the nodes, *i.e.* 1 to 8, except 3 and 4 which are inside the device. The states

across the heat exchanger are not measurable. By taking into consideration the assumptions described in Sections 2 and 3, we have in total 45 states of which 18 are assumed to be measurable and 27 are not measurable and are estimated by using the UKF. Accordingly, $n_x = 45$, $n_v = 45$ and $n_n = 18$ as in (8). Additive process and measurement noise was assumed to be Gaussian with zero mean and covariance.

$$\begin{cases} P_v = & \text{diag}\{v_T\mathbf{I},\ v_d\mathbf{I},\ v_{\dot{m}}\mathbf{I}\} \\ P_n = & \text{diag}\{w_T\mathbf{I},\ w_d\mathbf{I},\ w_{\dot{m}}\mathbf{I}\}\ , \end{cases} \tag{10}$$

In Equation 10 $v_T = v_d = 0.01$, $v_{\dot{m}} = 1\times10^{-5}$, $w_T = w_d = w_{\dot{m}} = 0.1$, with identity matrices of appropriate dimensions according to the number of states and measurements. It is thus assumed that the observation noise is considerably larger than the process noise, and this case is considered to validate the robustness of the UKF with respect to noise. The parameters for the UKF where chosen as $\alpha = 1$, $\beta = 2$ and $\kappa = 0$. The simulation was run for $t \in [1, 400]$ seconds in steps of 1 second, where an abrupt change in the gas turbine load from 100% to 90% occurred at $t = 50$ seconds. The initial values where selected as

$$\begin{cases} \mathbf{x}_T(t_0) = & [323.1\ 324.9\ 377.6\ 392.1 \\ & 406.3\ 420.7\ 435\ 449.6\ 464.5\ 479.3 \\ & 481.1\ 481.1\ 507.4\ 410.9\ 340.9]^\top \\ \mathbf{x}_{\dot{m}}(t_0) = & 44.4\mathbf{I}_{15\times1} \\ \mathbf{x}_d(t_0) = & [714.9\ 716.5\ 658.9\ 641.2 \\ & 622.7\ 602.5\ 580.5\ 555\ 523.9\ 481.5 \\ & 211.8\ 111.2\ 71.2\ 2.7\ 2.7]^\top. \end{cases} \tag{11}$$

The simulation results for the scenario described above are shown in Figures 4–6(b). Figure 4 illustrates the temperature measurements with noise, the real and the UKF estimated temperatures at the inlet of the once-through boiler. Given the magnitude of the measurement noise, it is demonstrated that the UKF is capable of estimating the real temperature. Figure 5 shows the real and the UKF estimated temperatures in an intermediate section (fifth cell) of the heat exchanger where the UKF attempts to evaluate the real temperature with no measurements available. Figures 6(a) and 6(b) show the real and UKF estimated temperature distribution over the length of the once-through boiler at $t = 40$ and $t = 200$ seconds. To be noted that the temperature distribution is similar to that occurring between nodes 3 and 6 in Figure 2. Consequently, we can conclude that the UKF can reliably reconstruct the internal temperature distribution with no measurements available. Thus it has been shown that the UKF is applicable to monitor the condition of the heat exchanger.

To test the robustness of the UKF with respect to initial conditions a new simulation was performed where the initial temperature estimates were different from the actual values. The

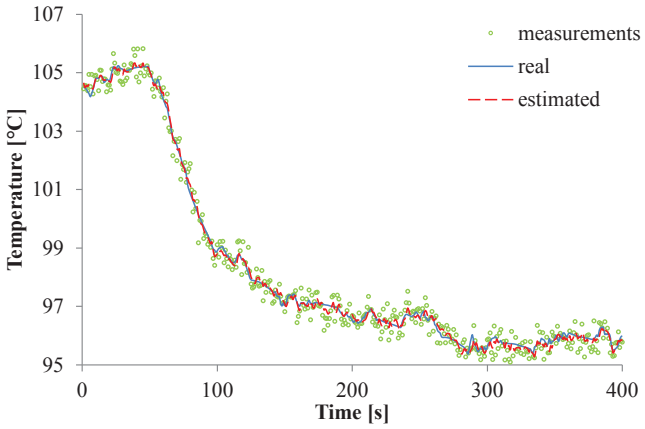


Figure 4. Real, estimated and measured temperature profiles at the inlet of the once-through boiler.

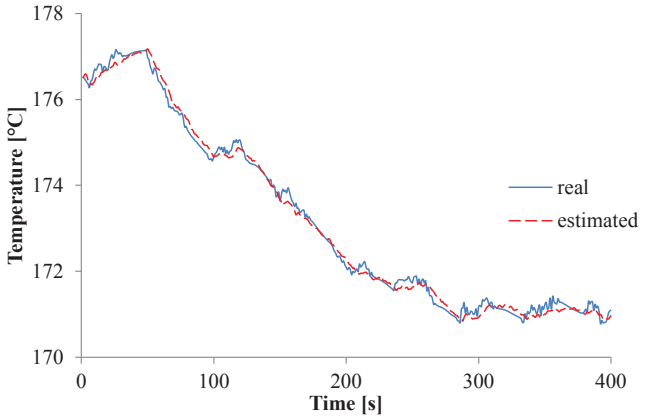


Figure 5. Real and estimated temperature profiles at an intermediate section of the once-through boiler.

new simulation was carried out so that all the first temperature estimates to the UKF are 3°C higher than the actual initial values. As the purpose of this simulation was to test the robustness with respect to initial state estimation, no change in the gas turbine load is applied. Otherwise the simulation set-up is similar to the previous case. The new results are shown in Figures 7–8. Figure 7 reports the temperature measurements with noise, the real and the UKF estimated temperatures at the inlet of the heat exchanger. Given the magnitude of the erroneous initial condition, it is demonstrated that the UKF estimate of the temperature is able to converge to the real temperature within the first time step. Figure 8 shows the real and the estimated temperatures at an intermediate section (fifth cell) of the once-through boiler. It can be observed that the UKF projection converged to the real temperature despite the lack of available measurements. Moreover, the estimate converged to the real temperature within approximately 100 seconds which is a reasonable time frame for this type of system.

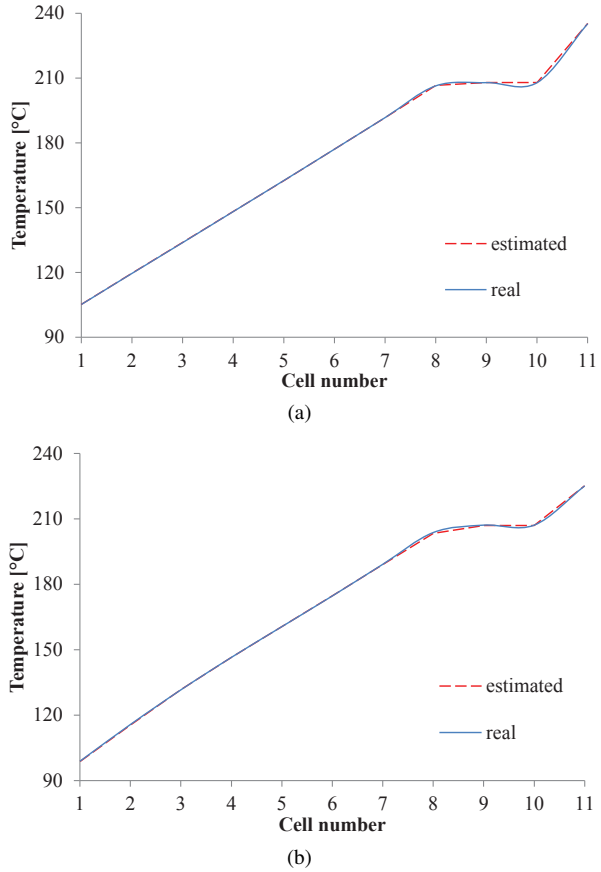


Figure 6. Real and estimated temperature distribution along the cells of the once-through boiler. 6(a) at time $t = 40$ seconds and 6(b) at time $t = 200$ seconds.

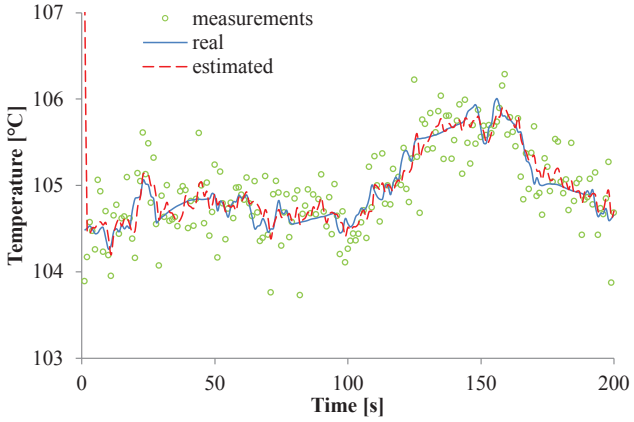


Figure 7. Real, estimated and measured temperature profiles at the inlet of the once-through boiler. The initial estimated temperature is 3°C higher than the real temperature.

6. CONCLUSION

This paper analysed the use of the Unscented Kalman Filter to predict the temperature profile inside a once-through boiler serving an organic Rankine cycle turbogenerator. Simulation results demonstrate the stability of the UKF, even with aggressive additive Gaussian noise profiles for process and measurements, and for a heat exchanger discretized into a relatively large number of volumes with unmeasured states. Furthermore, it was observed that the estimated temperature converged to the real values in a reasonable time frame when relatively reasonable deviations in the initial guess for all nodes were applied.

Future work will focus on applying the UKF to the combined cycle unit consisting of the gas turbine and the ORC turbogenerator. The estimation results will be embedded in the control algorithm of the integrated system. Further work will be directed towards the implementation of UKF with constraints on the state variables applied to this integrated system.

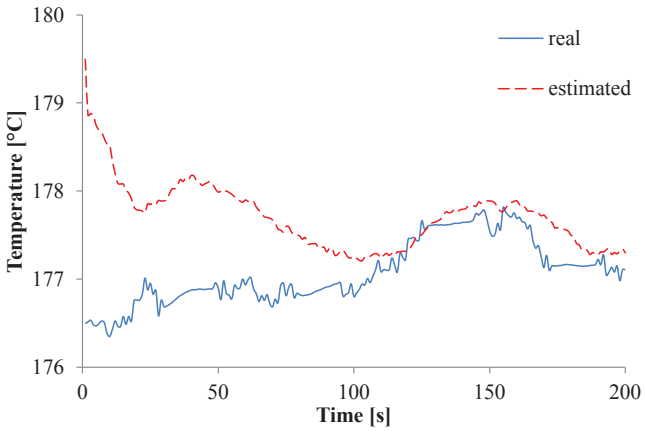


Figure 8. Real and estimated temperature profiles at an intermediate section of the once-through boiler. The initial estimated temperature is 3°C higher than the real temperature.

ACKNOWLEDGEMENT

The funding from the Norwegian Research Council through Petromaks with project number 203404/E30, which is lead by Teknova, is acknowledged.

REFERENCES

Bolland, O., Forde, M., & Hånde, B. (1996). Air bottoming cycle: use of gas turbine waste heat for power generation. *Journal of engineering for gas turbines and power*, *118*, 359-368.

Del Turco, P., Asti, A., Del Greco, A., Bacci, A., Landi, G., & Seghi, G. (2011, June). The ORegen waste heat recovery cycle: Reducing the CO_2 footprint by means of overall cycle efficiency improvement. In *Proceedings of ASME Turbo Expo 2011* (pp. 547–556). Vancouver,

Canada.

Haglind, F., & Elmegaard, B. (2009). Methodologies for predicting the part-load performance of aero-derivative gas turbines. *Energy*, *34*(10), 1484 - 1492.

Incropera, F. P., DeWitt, D. P., Bergman, T. L., & Lavine, A. S. (2007). *Fundamentals of heat and mass transfer* (6th ed.). Jefferson City, United States of America: John Wiley & Sons, Inc. (ISBN: 9780470501979)

Jonsson, G., & Palsson, O. P. (1994). An application of extended kalman filtering to heat exchanger models. *Journal of dynamic systems, measurement, and control*, *116*(2), 257–264.

Jonsson, G. R., Lalot, S., Palsson, O. P., & Desmet, B. (2007). Use of extended kalman filtering in detecting fouling in heat exchangers. *International journal of heat and mass transfer*, *50*(13), 2643–2655.

Julier, S. J., & Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Proceedings of signal processing, sensor fusion, and target recognition VI* (pp. 182–193).

Kandepu, R., Foss, B., & Imsland, L. (2008). Applying the unscented kalman filter for nonlinear state estimation. *Journal of Process Control*, *18*(7), 753–768.

Kloster, P. (1999, 7-9 September). Energy optimization on offshore installations with emphasis on offshore combined cycle plants. In *Offshore europe conference* (p. 1-9). Aberdeen, Great Britain: Society of Petroleum Engineers.

Loparo, K., Buchner, M., & Vasudeva, K. (1991). Leak detection in an experimental heat exchanger process: a multiple model approach. *Automatic Control, IEEE Transactions on*, *36*(2), 167–177.

Nguyen, T.-V., Pierobon, L., Elmegaard, B., Haglind, F., Breuhaus, P., & Voldsund, M. (2013). Exergetic assessment of energy systems on North Sea oil and gas platforms. *Energy*, *62*(0), 23 - 36.

Pierobon, L., Casati, E., Casella, F., Haglind, F., & Colonna, P. (2014). Design methodology for flexible energy conversion systems accounting for dynamic performance. *Energy*, *68*, 667–679.

Pierobon, L., Haglind, F., Kandepu, R., Fermi, A., & Rossetti, N. (2013, November). Technologies for waste heat recovery in off-shore applications. In *Proceedings of ASME 2013 International Mechanical Engineering Congress & Exposition* (p. 1-10). San Diego, California.

Pierobon, L., Nguyen, T.-V., Larsen, U., Haglind, F., & Elmegaard, B. (2013). Multi-objective optimization of organic Rankine cycles for waste heat recovery: Application in an offshore platform. *Energy*, *58*, 538–549.

Quoilin, S., Broek, M. V. D., Declaye, S., Dewallef, P., & Lemort, V. (2013). Techno-economic survey of organic Rankine cycle (ORC) systems. *Renewable and Sustainable Energy Reviews*, *22*, 168–186.

Schobeiri, M. (2005). *Turbomachinery flow physics and dynamic performance*. Berlin, Germany: Springer Berlin. (ISBN: 9783540223689)

Stodola, A. (1922). *Dampf- und gasturbinen: Mit einem anhang über die aussichten der wärmekraftmaschinen*. Berlin, Germany: Springer Berlin. (ISBN: 7352997563)

Veres, J. P. (1994). *Centrifugal and axial pump design and off-design performance prediction* (Tech. Rep.). Sunnyvale, United States of America: NASA. (Technical Memorandum 106745)

Wan, E. A., & Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive systems for signal processing, communications, and control symposium* (pp. 153–158).

BIOGRAPHIES

L. Pierobon is a PhD candidate at the Technical University of Denmark (DTU). He obtained his BSc and MSc degrees in Energy Engineering from University of Padova, Italy in 2008 and 2010, respectively. He is currently involved in the EOOS project financed by the Norwegian research council; the primary scope is to optimize the energy production in offshore platforms.

R. Schlanbusch received his BSc and MSc degrees in Space Technology from Narvik University College (NUC), Narvik, Norway, in 2005 and 2007 respectively, and PhD degree in Engineering Cybernetics from NTNU, Trondheim, Norway in 2012. He currently holds a position as senior researcher at Teknova, Norway, and a II position as associate professor at department of Technology, NUC, Norway.

R. Kandepu received his Masters in Control and Instrumentation from Indian Institute of Technology (IIT), Delhi, India in 2003. He received his PhD degree in Engineering Cybernetics from NTNU, Trondheim, Norway in 2007. He currently holds a position as senior researcher at Teknova, Norway.

F. Haglind received his MSc degree from Lund University, Sweden, in Mechanical Engineering in 1999 and his PhD degree from Cranfield University, UK, in Mechanical Engineering in 2005. Currently he holds a position as Associate Professor at Technical University of Denmark, Department of Mechanical Engineering. His primary research field is power plant engineering, with focus on low-temperature power cycles and gas turbines/combined cycles. He teaches in fundamental and applied engineering thermodynamics.

A Probabilistic Approach for Reliability and Life Prediction of Electronics in Drilling and Evaluation Tools

Amit A. Kale[1], Katrina Carter-Journet[2], Troy A. Falgout[3], Ludger Heuermann-Kuehn[4], Derick Zurcher[5]

[1,2,3,4,5]*Baker Hughes Incorporated, Houston, Texas, 77379,USA*

amit.kale@bakerhughes.com
katrina.carter-journet@bakerhughes.com
ludger.heuermann@bakerhughes.com
derick.zurcher@bakerhughes.com
troy.falgout@bakerhughes.com

ABSTRACT

The capability to predict performance and lifetime of drilling electronics is the key to preventing costly downhole tool failures and ensuring success of any drilling operation. Drilling electronics operate under extremely harsh downhole environments with temperatures beyond 150C and vibration levels exceeding 15g. In addition to temperature and vibration, there are several factors affecting electronic reliability that have high uncertainty and cannot be accurately measured. There is a growing trend in the oil and gas industry to drill faster and operate at higher temperatures and pressures, forcing tools to operate beyond design specifications. This has resulted in increased failure rate leading to higher maintenance costs and system downtime for drilling operators as well as service providers. This paper develops a methodology to estimate the life of drilling electronics by using operational data, drilling dynamics and historical maintenance information. The methodology combines parameter estimation techniques, statistical reliability analysis and Bayesian math in a probabilistic framework. Parameter estimation is used to calibrate statistical equations to field data and probabilistic analysis is used to obtain the likelihood of failure. In the paper, the model parameters are represented as random variables, each with a probability distribution. Drilling electronics under downhole conditions can have several failure modes and each failure mode can be caused by the interaction of several variables. When information on each failure mechanism is not readily available, the failure is expressed in terms of several candidate models. Bayesian updating is used to incorporate real time operational history for a specific part and select the most accurate failure model for that part. Tis is for the first time, a systematic approach is developed for predicting the life of electronics in downhole drilling environments using statistical modeling and probabilistic methods on life cycle history and operational data from the field.

Amit Kale et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Drilling and evaluation operations are becoming faster, more accurate and safer, thanks to modern electronics that enable measurements, storage and transmission of information in real time. Transmitting information in real time makes it possible to evaluate properties of earth's formation while drilling and enable directional drillers to steer wells towards target zones more efficiently. The reliability of electronic printed circuit board assemblies (PCBAs) in the bottomhole assembly (BHA) is the key to the success of any drilling operation. Drilling electronics operate in extremely harsh downhole environments with temperatures exceeding 150C, shock and vibration levels exceeding 15g. The impact of temperature, shock and vibration on the life of electronics is described by Barker et al. (1992), Duffek (2004), Garvey et al. (2009), Gingerich et al. (1999), Lall et al. (2005, 2007), Mirgkizoudi et al. (2010), Pecht et al. (1999), Vichare (2006), Vijayaragavan (2003), Wassell & Stroehlein (2010), White & Bernstein (2008). Other factors like power cycles, thermal ramp rates, electrical overstress, mechanical stress and manufacturing defects impact reliability of tools, but the factors cannot be accurately measured in downhole drilling environments and encompass high uncertainty. These factors can act alone or interact with each other to produce several degradation mechanisms that can cause failure. For example, Mirgkizoudi et al. (2010) demonstrated through tests that there is significant difference between the lives of electronic components subjected to thermal testing with vibration as compared to those with pure thermal loading. Failure of electronics because of fatigue, corrosion, electromigration, filament formation and dielectric breakdown has been

established by the scientific community (e.g. Barker et al. 1992, Duffek 2004, Gingerich et al. 1999, Lall et al. (2005, 2007), and Pecht et al. 1999). Typical PCBAs used in the drilling industry are multiscale devices made from several components. The geometric dimensions of individual components may vary from nanometers to inches. This difference creates significant challenges in developing a predictive model for failure because individual components on a PCBA may fail by many failure modes based on the operating environmental conditions. Furthermore, diagnosis of faults and indicators of failure is difficult because degradation of individual components may not lead to a measurable loss of electrical function up until imminent failure. There is growing interest in the area of health prognostics for electronic components through the use of physics based models, operating data from fielded products, design qualification testing and in-service inspections (e.g. Pecht et al., 1999, Vichare 2006, and Garvey et al., 2009) The main drivers behind the efforts are preventing failure and system downtime, reducing costs of repair and maintenance, and supporting new product improvements. A discussion on state of the art techniques in prognostics and health management of electronics can be found in Pecht et al. (1999) and Vichare (2006).

The method of measuring failure precursors as indicators of impending failure is based on the hypothesis that degraded circuit boards produce significantly different signatures from defect free boards. Failure precursors are measurable indicators that can be correlated with subsequent part failures. Failure indicators for electronics like shifts and variation in temperature, voltage, current, surface insulation resistance and impedance have been proposed by Born & Boenning (1989) and Pecht et al. (1997, 1999). Another area of research in electronics prognostics and health management (PHM) is usage of sacrificial circuits like fuses, canaries, circuit breakers and self-diagnostics sensors for detecting if the device is operating outside of design limits. These devices are mounted along with the main electronic component but have accelerated failure rates to provide advance warning of failure (e.g. Mishra & Pecht 2002, and Ridgetop Semiconductor Sentinel Silicon report 2004).

The physics of failure (PoF) based approach for life prediction uses modeling and simulation to relate the fundamental physical and chemical behavior of materials to the surrounding environment and applied loads. The PoF based modeling process starts by exposing the product to the highly accelerated life test (HALT) and highly accelerated stress test (HAST) to find the significant modes and root cause of failure. Next, the governing equations of the failure mechanisms are combined with the data gathered from acceleration tests using statistical distributions. The PoF approach has been successfully applied to understand system performance, identify weak links and root cause of failure so that they can be mitigated before the product is launched. Chatterjee et al. (2012) gives a historical perspective of the evolution of the physics of failure approach. White & Bernstein (2008) present the state of the art methods for PoF modeling. Finite element analysis was used to model fatigue damage growth during cyclic loading (thermal, mechanical and combination of both) by Barker et al. (1992), Bailey et al. (2007), Dasgupta (1993), Duffek (2004), Shinohara & Yu (2010), and Vijayaragavan (2003). Material modeling to predict degradation of solder joints in the circuit board as results of thermo mechanical fatigue was developed by Nasser & Curtin (2006). Lall et al. (2007) used experimental tests in combination with finite element analysis to model solder joint failure from shock and vibration. Mirgkizoudi et al. (2010) developed a test plan to evaluate the reliability and service life of electronic components that are subject to a combination of mechanical, thermal, chemical or electrical inputs, and Wassell & Stroehlein (2010) use accelerated tests to derive accumulated damage models and failure thresholds as functions of vibration, shock levels, the number of shocks and the operating temperature. Young & Christou (1994) developed models for failure because of electromigration. The models obtained from accelerated tests are also widely used to estimate the life for fielded products by using the governing equation to scale accelerated test life to that under the actual operating environment in the field. However, such scaling is valid only if the following conditions are met (1) failure modes and mechanisms for accelerated stress levels are the same as those observed in the field and (2) variations of material properties with stress levels are incorporated in the governing equations. Because of these limitations, it has been shown for practical application that life obtained by scaling the highly accelerated life tests (HALT) and highly accelerated stress tests (HAST) is orders of magnitude different from those observed in actual field environments (e.g. Osterman 2001, Pecht (1997, 1999), and White & Bernstein 2008).

Field data driven methodologies for modeling time to failure have gained momentum because of the availability of large volumes of data and limitations of physics based methods to simulate actual operating environment in laboratory (e.g. Osterman, M., 2001 and Vichare 2006). This methods use operating environment measured in field, repair and maintenance information of fielded products in conjunction with statistical modeling to predict the life of parts in operation. For example, Hu et al. (1991) presented a probabilistic approach for predicting thermal fatigue life of wire bonding in microelectronics, and Vichare et al. (2007) developed an algorithm to extract load parameters necessary for assessing damage from commonly observed failure mechanisms in electronics. Sutherland et al. (2003) developed data mining methods and statistical approaches to obtain accurate life distribution for power plant maintenance optimization.

There is a growing trend in the oil and gas industry to drill faster and operate at higher temperatures and mechanical loads, forcing tools to operate beyond design limits. The capability to predict performance and life of drilling electronics is critical to preventing costly downhole tool failures and reducing cost of maintenance. This paper presents a systemic approach for deriving and updating models for time to failure of PCBAs used in drilling and evaluation tools using field data. The methodology combines parameter estimation techniques, statistical reliability analysis and Bayesian math in a probabilistic framework. Parameter estimation technique is used to calibrate statistical equations to field data and probabilistic analysis is used to obtain the likelihood of failure. The model parameters are represented as random variables with probability distribution. Drilling electronics within downhole conditions can have several failure modes and each failure mode can be caused by the interaction of several variables. When information on each failure mechanism is not available in real time, the failure is expressed in terms of several candidate models. Bayesian updating is used to incorporate the operational load history for a specific part and selecting the most accurate failure model for the part. Results presented in the paper show that the life of electronic assemblies used in drilling and evaluations can be predicted accurately by using the probabilistic model and incorporating operational effects. Interaction between different factors causes the components to degrade faster than individual factors acting alone.

2. OPTIMAL MAINTENANCE PLANNING

The framework for lifecycle management, optimal operations, repair and maintenance planning of drilling systems requires databases to record equipment lifecycle history, environment and operations data, telemetry and communication systems, sensor and measurement systems and algorithms for predicting performance and consumed life. Developing an optimal maintenance strategy requires the knowledge of component life as a function of usage. Predicting component life accurately requires knowledge of engineering design, physics of component behavior under operating loads, data from qualification tests, operating mission of fielded products and indicators of degradation of part life from inspection and maintenance shops. The information can be used in physics based or statistical data driven models (or a combination of both) to predict part life and risk of failure as a function of usage. Once accurate life models are developed, cost factors, performance and reliability targets can be incorporated to optimize maintenance plans for minimum life cycle cost. In field operations, life extension can be achieved by derating the mission (e.g. lowering rotational speed of drill to reduce impact of vibration induced damage on BHA components) so that parts degrade slower. Cost of repair and maintenance can be lowered by using a risk based maintenance level. For example, tools with low risk of failure can be given a quick turnaround, medium risk entails partial disassembly and inspection, and high risk tools require full piece part level disassembly and inspection. The goal of this method is to enable reliability and maintenance personnel to schedule timely maintenance and prevent costly downhole tool failures. Fig. 1 shows a high level overview of data, methods and decision process for optimizing operations and maintenance plans.

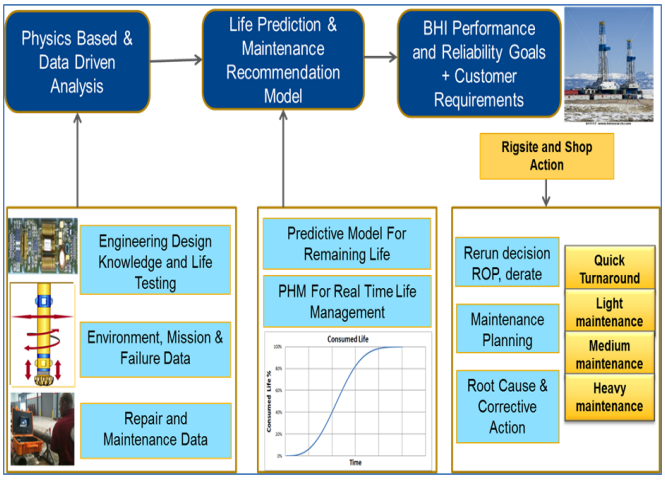


Figure 1. Methodology for optimal operations and life management of parts.

This paper develops a framework to provide advance warning of impending failure so that high risk components can be retired. The remainder of the paper focuses on algorithms to estimate part life using data from field and maintenance shops. Section 3 gives an overview of parts in the bottomhole assembly (BHA) for which reliability models are developed. Section 4 describes the algorithms used to analyze field data and develop mathematical models for time to failure. Section 5 describes the methodology to use load history from each drilling mission (also known as a "run") to update model weights and predict part life. Section 6 presents results for fielded component and Section 7 concludes the paper with a summary and future work.

3. DESIGN OF BOTTOM HOLE ASSEMBLY

A typical drilling system comprises a drill bit, bottomhole assembly (BHA); drill pipes and rig (Fig. 2**)**. The drill bit is a rotary cutting tool that cuts through the earth's formation; the drilling rig is a structure on the surface that houses equipment, the drill pipes provide the required extension to reach a target depth and the bottomhole assembly (BHA) is a structure that houses drill collars, reamers, steering system and electronic components. The focus of the report is predicting life of electronic components in BHA of the AutoTrakG3 line of product manufactured by Baker Hughes Incorporated. A typical AutoTrakG3 contains three modules, namely (1) the

AutoTrak steering system (ASS) that provides the necessary drive to steer the bit (2) OnTrak sensor assembly contains the electronics used for measurement while drilling (MWD) and logging while drilling (LWD). The OnTrak tool takes measurements like resistivity, gamma ray, pressure and vibration. (3) Bi-directional communication and power module (BCPM). This module sends and receives data to and from the surface, enabling drillers to monitor drilling operations in real time and make adjustments when necessary. The BCPM also delivers power required by the other modules in BHA. The three assemblies have components that are critical to the drilling and evaluation operation. Failure of the components can lead to the loss of functionality and cause trip for failure which can cost several millions of dollars. The paper focuses on developing predictive life models of several such components in the drilling system.

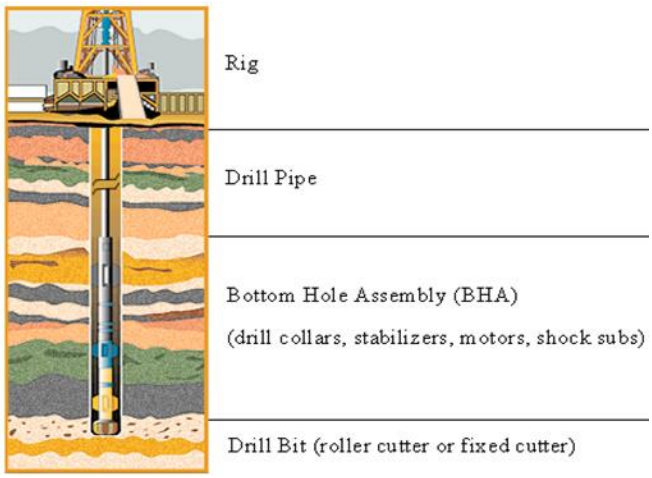


Figure 2. Illustration of drilling system.

4. FIELD DATA ANALYTICS

Developing field data driven models for life of electronic assemblies in drilling operations is challenging for two reasons. First, not all of the factors impacting component life can be measured in real time, and second, the data that can be measured has errors and noise because of limitations of the measurement system and human factors. This paper presents method to calculate the reliability of components that have been operated at varying stress level because of temperature and mechanical loads such as that caused due to shock and vibrations. The Maintenance and Performance System (MaPS^{TM}) is a state of the art database developed by Baker Hughes Incorporated to track equipment lifecycle data. Information related to operations, failure, repair and maintenance is stored for serialized parts. The downhole environment data like temperature, vibration, pressure and power cycles is also maintained in the MaPS database. The magnitude and cyclic variation of temperature can cause solder joint fatigue failure in electronic circuit components, chip delamination, corrosion, electro migration, diffusion voids and dielectric breakdown. Extreme vibrations influence the life of electronic components in the BHA. There are three principal modes of vibration: (1) axial vibration along the tool axis can cause damage to seal faces of modular connections, stabilizers and, in severe cases, can lead to buckling fatigue. Axial vibration is responsible for low rates of penetration and reduced efficiency, (2) lateral vibrations occur transversely to the tool axis. Historically, they are the most destructive type of vibrations and constant exposure to lateral vibrations can cause damage to tool electronics. Constant lateral shocks damage the tool body as well as greatly reduce drilling efficiency, (3) stick slip is a rotational phenomenon that occurs because of twisting of the drill string. Twisting can occur when the bit gets stuck downhole while the motor continues to turn the drill string. When the bit is free, the torsional energy stored in the drill string is released, causing the BHA to spin in the opposite direction. Stick slip can lead to material fatigue and physical damage to the tool and electronics. Figure 3 shows the three vibration modes.

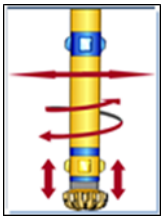


Figure 3. Vibration modes in drill string.

4.1. Consolidating Life Cycle Data

An important first step in developing a life model is to collect life cycle history for each part. Each serialized part undergoes one of three maintenance actions during its lifecycle: (1) repairs, which involve replacing damaged components on a PCBA, (2) revision upgrades which may include repairs and/or firmware updates, (3) scrapped because of failure or as a preventive measure. To accurately capture the life cycle of a part, the accumulated temperature and vibration hours for each serialized part are retrieved from MaPS database and grouped using the steps described in Table 1. The purpose of the steps described in Table 1 is to group the data into buckets that have three common characteristics, namely revision id flag, repair flag, and revision upgrade flag. Data in each bucket encompasses the same value for the three flags and any two buckets have at least one flag different between them. For example, the bucket in which the three flags are [“A”, N, N] implies that parts in that bucket are revision “A”, they have never been repaired and never received a revision upgrade. Another bucket with flags [“A”, N, Y] implies that parts in that bucket have never been repaired and have been upgraded to revision “A” from an older revision. A bucket with flags [“A”, Y, Y] implies that all parts in that bucket have been

repaired and have been upgraded to revision "A" from an older revision.

Table 1. Process to group part life cycle data for failures, suspensions, repairs and revision upgrades.

(1) Find all the serial numbers of a given part number in the database
(2) Select a serial number and look up mission profile for that serial number starting with installation date
(3) Accumulate drilling hours, circulating hours and the operating environment variable (temperature, vibration, rotational speed (rpm), distance drilled) etc. for each run; store the accumulated data in a record with index *i*. Store the revision id flag, repair flag (*Y/N*), revision upgrade flag (*Y/N*), and failure/suspension flag (*F/S*)
(4) Check if the part underwent one of the following actions after the run (a) failed and scrapped, (b) failed and repaired to put back in service (c) upgraded to next revision (d) repaired to put back in service (e) scrapped because of preventive maintenance. If any of the above is true, then label the i^{th} record flag appropriately. Create a new record *i*+1 and go to step 3. If none of steps (a)–(d) happened, continue to accumulate the fields for the i^{th} record in step 3
(5) Check if all the runs have been accounted for the serial number. If no, go to step 3; otherwise, create a new record for a new serial number

It is important to make the distinction between revision upgrade and repair because not all revision upgrades lead to life extension (for example, if only firmware is changed in revision upgrade). Grouped data is filtered for outliers and weighted before building a life model using an algorithm described in the next section.

4.2. Iteratively Reweighted Maximum Likelihood Algorithm

The life cycle data for parts recorded in the maintenance database is large and complex because each part has several hundred serial numbers and each serial number has the operating history for several drilling runs. Like any other physical experiment, data can have errors or noise because of human factors and flaws in the measurement system. The impact of outliers on the quality of the predictive model can be minimized by optimally weighting the life cycle data. Outlier identification is done by first removing data points that lead to constraint violation in the estimation process. The likelihood equation is subjected to constraint that $\alpha_0 > 0$ *and* $\alpha_{1\ldots}\alpha_n \leq 0$ in Eq. A-1, A-5 and A-8. The inclusion of these constraints implies that life decreases with increase in stress level due to temperature and vibration. Next, iteratively reweighted maximum likelihood estimation (IRMLE) technique was developed to determine the optimal weight of each data point in the life cycle data. Unlike conventional likelihood maximization procedure where all points are weighted equally, the new technique iteratively maximizes the weighted likelihood function of life data until the quality of model shows no further improvement. Iteratively reweighted maximum likelihood estimation procedures assign weight that is inversely proportional to the log-likelihood of the data point, so that points with lower log-likelihood are weighted less than points with higher log-likelihood. Eventually, the model moves away from outliers. The procedure can be summarized in steps (1)-(4). The symbols used in these steps have the following description.

T is temperature, *L* is lateral vibration, *S* is stick slip or rotational vibration, *RPM* is revolutions per minute, α_0 is a constant term, $\alpha_{1\ldots}\alpha_n$ are coefficients on stress variables in the life equation (e.g. Eq. A-1, A-5 and A-8), $w^i_{updated}$ is the model weight, symbol £ is likelihood of i^{th} data point.

(1) Select $\bar{\boldsymbol{X}} = \{T, L, S, RPM, LT, ST, LS, SRPM\}$ for modeling characteristic life function described in Appendix A.

(2) Maximize weighted sum of likelihood of failure and suspension data to estimate the mean and variance of parameters of the characteristic life function (e.g. Eq. (A-1) α_0, $\alpha_{1\ldots}\alpha_n$). The initial weight of each data point is unity. The maximization of likelihood equation is subjected to constraint that $\alpha_0 > 0$ and $\alpha_{1\ldots}\alpha_n \leq 0$.

(3) Compute the value of likelihood of each data point at the values of α*'s* estimated in step 2. Compute the mean and standard deviation of likelihood, $\mathcal{L}_{mean}$ and $\mathcal{L}_{stdev}$. The updated weight $w^i_{updated}$ of i_{th} data point is given by

$$w^i_{updated} = \frac{Total\ number\ of\ data\ points}{\Sigma \frac{\mathcal{L}^i}{\mathcal{L}_{stdev}}} \frac{\mathcal{L}^i}{\mathcal{L}_{stdev}} \qquad (1)$$

(4) Iterate step (2) – (3) with updated model weights until the sum of likelihood has converged within a specified tolerance (10^{-6} used in this paper).

In principle the IRMLE technique is similar to the iteratively reweighted least squares (IRLS) except that in IRMLE, the weighted sum of likelihood is maximized, whereas in IRLS the weighted sum of squares of difference between data and model response is minimized. The IRMLE algorithm is used to build transfer function for time to failure as a function of the operating mission for a serialized part. One of the challenges in using this model to accurately estimate remaining life is that the operating environment is variable throughout the life of a component. This is overcome by updating the remaining life estimate after each drilling mission (life of a part can span several drilling missions and each mission may have different load history and hours). The application of this algorithm in identifying outliers is presented in Fig. A1 through Fig. A6 in Appendix A.

5. RELIABILITY ANALYSIS

Statistical models are extensively used in reliability and life data analysis to estimate time to failure of parts in operation. The models are either computational simulations or a set of mathematical equations that explain the general state of a system under the influence of load and time. Typically, a mathematical model is an approximation of the physical phenomena and rarely matches the field observations. However, for practical commercial application where the models are used in design and operation of a product, it is desirable to have a model that matches the field or experimental data closely. The process of determining the unknown model parameters by tuning the model to field data is called parameter estimation or model calibration. The model parameter usually represents quantities that have physical significance and are determined by imposing some constraints during the calibration process. The constraints require that the parameters being estimated must have minimum variance from using one set of data to the next and the estimated value is bound to the true value. A reliability model that best represents the life cycle of a component can be developed when sufficient amount of operation, failure, and repair and maintenance data is available. This section outlines the method for calibrating a mathematical model to field data and its subsequent application to predict remaining life and reliability using real time mission profile for a specific part.

5.1. Generating Best Fit Model

A typical time to failure model comprises a life distribution function to incorporate the statistical scatter in failure time and a characteristics life function (Appendix A) that describe a general relation between failure time and stress levels. In this work, the Weibull, lognormal and exponential distributions are used to build time to failure models. The life characteristic can be any life measure such as the mean, median or hazard rate that represents a bulk property of the distribution. The life characteristic is expressed as a function of stress (as shown in Appendix A). The unknown parameter of the composite model is determined by tuning the model equation to field data using the Iterative Maximum Likelihood Estimation technique. The method for deriving the model that best fits the field data is described in the following steps:

(1) Retrieve life cycle data from maintenance database and bucketize it using the method described in Section 4.1.
(2) Select a revision identifier, trial function for stress η_i and trial function for probability distribution f_j from Appendix A. Initialize trial functions, i=1, j=1.
(3) Calibrate the reliability model $f(t,x)_{ij}$ to the bucketed field data using IRMLE technique. Compute standard deviation in parameter estimates.
(4) Compute goodness of fit for model $f(t,x)_{ij}$ by evaluating prediction error sum of squares (PRESS[1]).
(5) Select new probability distribution and trial function by updating values of i and j and repeat steps (2) – (4) until all trial functions are evaluated.
(6) Generate pareto of the solution obtained from steps (1) – (5) with two objectives namely, goodness of fit and Euclidean norm [2] on coefficient of variation of parameter estimates.

The models generated by steps (1)-(4) yield pareto of competing solutions, some solutions are better in terms of cross validation error while others are better in terms of confidence in value of estimated model parameters (α's described in Appendix A). The time to failure for a part in operation is determined using the method described in the next section.

5.2. Model Selection and Updating Using Real Time Data

The best fit model is representative of a nominal [3] part. Drilling electronics under downhole conditions can fail because of several mechanisms that can be caused by the interaction of several variables (like temperature, vibration, and power cycles). The time to failure is expressed as weighted average of several competing models. Bayesian updating is used to select the most accurate failure model for a specific part by using the real time mission profile for that part. Bayesian updating provides a systematic process for incorporating real time operational data for model selection and updating. This section presents Bayesian formulation for updating probability of an event y based on recorded observations at time t (examples of observations include pass/fail event and mission profile parameters like temperature, lateral vibration, stick slip, etc.). More details on this formulation can be found in Zhang and Mahadevan, (2000). The symbol M_i is the i^{th} model, $p(M_i)$ [4] is the probability of i^{th} model and reflects the belief that the model is accurate for the specific part in operation, $p(y|\bar{x}_i, t, M_i)$ is the probability of observing an outcome y at time t using the i^{th} model, the vector $\bar{x}_i$ is a set of parameters estimated by the calibration procedure. The term $f(\bar{x}_i|M_i)$ is the joint probability density function of the parameters of i^{th} model.

[1] PRESS is adding the squared of difference between data and model prediction, where the model is constructed by excluding one data point and repeating this over all the data points.
[2] Euclidean norm of an n-dimensional vector space is given by the geometric distance from origin to a point x.
[3] A representative part that has a life equal to the average of several part produced using same manufacturing process and operating under same condition
[4] Note that $\sum p(M_i) = 1.0$

The event *y* is the state of the part at a time *t* that has one of the two values *z* = *pass* or *fail*.

$$p(y) = \sum_{i=1}^{m} p(M_i) \int_{x_i} p(y|\bar{x}_i, t, M_i) f(\bar{x}_i|M_i)\, d\bar{x}_i \qquad (2)$$

The prior probability $p(G_i)$ of the parameters of i^{th} model is given by Eq. (3).

$$p(G_i) = p(M_i) f(\bar{x}_i|M_i) \qquad (3)$$

$p(G_i)$ is the prior probability of $(M_i, \bar{x}_i)$ pair. The posterior probability after observing an outcome for $y=z$ is given using Bayes theorem in Eq. (4).

$$p(G_i|y=z) = p\big((M_i|y=z)\big) f(\bar{x}_i|M_i, y=z)$$

$$= \frac{p(y=z|G_i)p(M_i)f(\bar{x}_i|M_i)}{\sum_{i=1}^{m} p(M_i) \int_{x_i} p(y=z|\bar{x}_i, t, M_i) f(\bar{x}_i|M_i)\, d\bar{x}_i} \qquad (4)$$

Integrating over the probability distribution of $\bar{x}_i$ in Eq. (4), the posterior model weight of the i^{th} model after observing an outcome $y=z$ is given by Eq. (5).

$$p(M_i|y=z) = \frac{p(M_i) \int_{\bar{x}_i} p(y=z|G_i) f(\bar{x}_i|M_i) d\bar{x}_i}{\sum_{i=1}^{m} p(M_i) \int_{x_i} p(y=z|\bar{x}_i, t, M_i) f(\bar{x}_i|M_i) d\bar{x}_i} \qquad (5)$$

It is important to note that the time *t* used in Eq. (2) through Eq. (5) is not the failure time but it is the time at which an observation is made regarding the pass or fail state. The expected time to failure is obtained by weighted sum of time to failure predicted by each of the models as shown in Eq. (6).

$$tf_{predicted} = \sum_{i=1}^{m} p(M_i|y=z) \times tf_{M_i} \qquad (6)$$

Where $tf_{predicted}$ is the expected life of a part being modeled and tf_{M_i} is the life predicted by the i^{th} model whose probability distribution is given in Appendix A. Equation 6 is solved using the Monte Carlo simulation technique. For drilling tools, probability of failure greater than 10% is unacceptable. To estimates this probability accurately we use a sample size of 10,000[5] in Monte Carlo simulation.

6. RESULTS

The methodology developed in this paper is used to predict life of fielded electronic assemblies used in drilling and evaluation tools and advance warning of impending failure so that preventive maintenance can be scheduled. The life cycle data for a typical low voltage power supply (LVPS) modem used in drilling operations is shown in Fig. 4 for parts that failed in field and Fig. 5 for suspensions (i.e. parts that are operating in field.). The *x* axis on the plots represents the average temperature (lateral vibration, stick slip and interaction effects are shown in Fig. A1-Fig. A6 in Appendix A). The *y*-axis represents drilling hours. Each point on the figure is a unique serial number of the part and undergoes different mission profile during their life. The data shown in Fig. 4 is derived from the failure of parts in operation that are root caused and Fig. 5 shows data for parts that are either currently being operated or those that are retired for precautionary measures.

Fig. 4 and 5 show field data with scatter and noise. As such, errors and noise cannot be totally eliminated and are part of field data because of limitations of the measurement system and human factors. The methodology developed in the paper is used to reduce the scatter in the life prediction by incorporating the cumulative effect of temperature, vibration and their interaction on life consumption. The IRMLE algorithm described in Section 4.2 is applied to the data in Fig. 4 and Fig. 5 and the outliers (shown in red dots) are identified by the algorithm. The data in Fig. 4 and Fig. A1 through Fig. A3 shows that temperature and vibration have a detrimental effect on life.

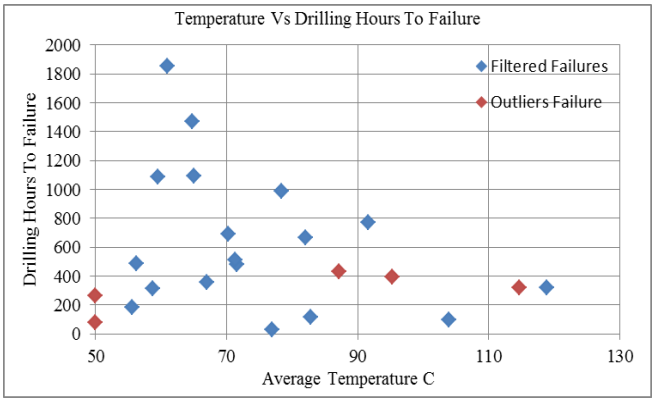


Figure 4. Time to failure vs. temperature severity for fielded LVPS modem serialized parts.

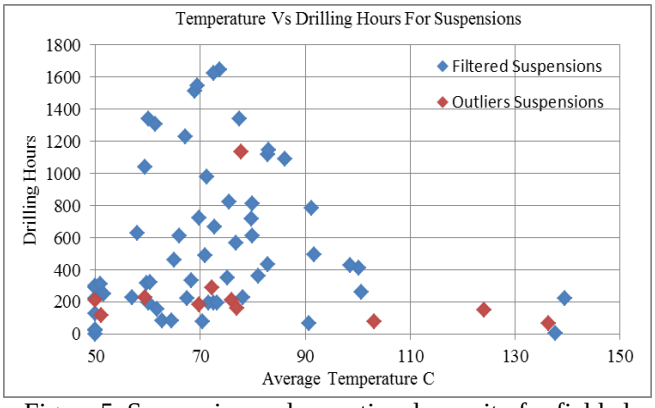


Figure 5. Suspension and operational severity for fielded LVPS modem serialized parts.

[5] The standard deviation in probability calculated by Monte Carlo integration is given by $\sqrt{\frac{p(1-p)}{10{,}000}}$. For a target probability of 50% the standard deviation is 0.005. Hence 10,000 samples are sufficient to estimate probabilities level of interest in this paper.

Table 2 show the parameters of the time to failure model built from the data in Fig. 4 and 5. The best fit model is a Weibull distribution with a characteristic life function whose parameters are α and β. The models are generated using the best fit procedure described in Section 5. The values in parenthesis are the mean and standard deviation of the parameter estimates. Each of the models in Table 2 is comparable in terms of likelihood value and confidence level in coefficients. Model M_1 shows the interaction of temperature and lateral are significant factors affecting the life of the part; model M_2 shows the temperature by itself is significant; and model M_3 shows the temperature plus interaction of temperature and stick slip are significant factors.

Table 2. Competing Weibull models for time to failure of apart as a function of operating stress.

Parameter	M_1	M_2	M_3
$P(M_i)$	0.29	0.40	0.31
$\alpha_0\ (\mu, \sigma)$	(7.5, 0.07)	(8.0 0.1)	(8.6, 0.1)
$T, \alpha_1\ (\mu, \sigma)$	**0**	(-10.3, 0.7)	(-7.9, 0.5)
$S\times L, \alpha_2\ (\mu, \sigma)$	**0**	**0**	(-43.8, 3.1)
$T\times L,\ \alpha_3\ (\mu, \sigma)$	(-39.3, 2.5)	**0**	**0**
$\beta(\mu, \sigma)$	(1.6, 0.08)	(1.7, 0.07)	(1.8, 0.05)

The models in Table 2 represent failure time for a nominal part representative of the population. To obtain an individual part specific prediction, the time to failure is expressed as a weighted sum of failure times from each of the models using the operational history from each run of that specific part and adjusting the relative contribution of each model using the Bayesian formulation in Section 5.2. An example is shown for predicting the time to failure for a single part in operation. Table 3 shows the load history on an LVPS modem operated for 1000 drilling hours at varying levels of temperature and vibration. The first column of Table 3 shows the run number which represents the mission between the start and stop of the drilling operation; the second column shows the average temperature for the run; the third column shows the average lateral vibration level for the run; and the fourth column shows the average torsional vibration level. The lateral and stick slip vibrations (reported as root mean square in units of acceleration because of gravity g) are measured by accelerometers placed in the drilling assembly. The algorithm described in Section 5 is applied to the operational history after each drilling mission (referred as a "run"). Starting with an equal model weight of 0.33 for the three models, the life prediction and model weight is updated after each run to obtain a more accurate estimate of remaining life after each run (using Eq. 3 through Eq. 6). The final value of model weights prior to the eighteenth run is shown in second row of Table 2 for each of the three candidate model.

The life expectancy predicted by Eq. 6 (shown in Table 2) and the actual hours accumulated on the part after each drilling run and the operating environment is shown in Fig. 6 and Table 3. Figure 6 shows the true remaining useful life (RUL) and 95 percent confidence bounds on predicted life. It can be seen that the true RUL is bounded between the predicted 95% confidence interval. This interval represents statistical variation in part life of the population of identical parts subjected to same load history. The variation is caused by defects in manufacturing, limitations of the measurement system and human factors that are unknown or cannot be modeled. The purple diamonds represent the actual RUL on the part. Fig. 6 shows during the early part of the part life cycle, the life expectancy is high, but with usage and application of operating loads, the accumulated hours begin falling within the range of variation of expected life. At that point, the component is retired to prevent downhole tool failure. The part failed during the nineteenth drilling run. In retrospect, the model accurately predicted impending failure when it showed that the part was at high risk (>75% risk of failure) from the seventeenth run and should have been retired at that time.

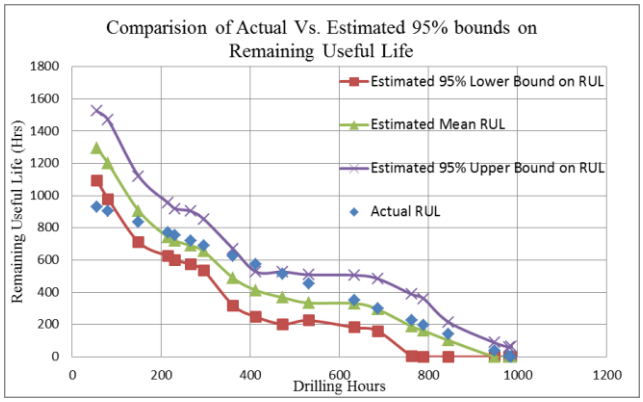


Figure 6. Predicted life vs. actual drilling hours after each run for LVPS modem.

Fig. 6 shows that the expected life of a part can increase or decrease with each run and are not a constant number (because expected life is a function of usage). Table 3 illustrates the concept where the average value of operational temperature and vibration over all the previous runs is calculated in columns two through four. The first run is the least severe and has the highest life expectancy. In subsequent runs, the life expectancy reduces as the severity of operation increases as shown by the values of temperature, lateral and stick slip vibrations. The trend continues until the ninth run, after which the operational severity starts reducing, leading to higher life expectancy until the thirteenth run. In summary, the life expectancy can vary through the operation depending on the severity of operating environment.

Table 3. Average operating environment and risk of failure after each drilling mission (run) during life of a part

Run No.	Average Temperature C	Average Lateral (g_RMS)	Average StickSlip (g_RMS)	DrillHrs [h]	Risk
1	57.6	1.6	0.2	55.3	0.00
2	63.8	1.5	0.1	80.8	0.00
3	57.6	1.3	0.3	149.2	0.00
4	71.9	1.1	0.2	215.4	0.00
5	74.9	1.1	0.2	231.0	0.00
6	72.0	1.1	0.2	266.1	0.00
7	70.1	1.1	0.2	295.1	0.00
8	77.3	1.0	0.3	361.4	0.00
9	81.8	0.9	0.3	412.6	0.00
10	78.9	0.9	0.3	472.6	0.00
11	76.5	0.8	0.3	530.6	0.00
12	73.0	0.9	0.2	633.8	0.00
13	71.2	0.9	0.2	686.4	0.00
14	71.7	0.9	0.3	761.5	0.00
15	73.3	0.9	0.3	788.5	0.03
16	75.5	0.9	0.2	844.9	0.25
17	79.6	0.9	0.2	948.0	0.85
18	78.6	0.9	0.2	981.0	0.90
19	78.4	0.9	0.2	986.0	0.87

7. CONCLUSIONS

The paper presents a generic methodology to predict the life of electronic components used in drilling and evaluation tools. Statistical modeling techniques are used to derive best fit mathematical equations for durability of parts from field data. The method is applied to predict life of electronic printed circuit boards (PCBAs) and retire high risk components. The key challenges associated with developing durability models for PCBAs in drilling environment are:

(a) Life of parts is impacted by several factors, not all which can be measured accurately because of limitations of measurement systems and human factors.
(b) Field data may have noise and errors that may affect the quality of predictive model.
(c) Statistical model do not incorporate physics of degradation and may not be applicable for all failure mechanisms.

The methodology addresses the aforementioned challenges for the first time vis-à-vis application to lifing parts operating in downhole drilling environments. The key features of the analysis methodology include:

(a) Algorithm to determine life from cumulative damage over time and the best-fit mathematical model using a combination of statistical distribution and characteristic life function.
(b) Clustering mechanism to group parts life cycle data by upgrades, repair, failures and suspensions.
(c) A pattern search and outlier detection algorithm to identify data from a physical degradation trend.
(d) Iteratively reweighted maximum likelihood estimation method to determine optimal weights of data points.
(e) A Bayesian model selection technique to incorporate part specific operational history to obtain improved accuracy in life prediction.

Future work will focus on improving model predictions by using additional environment variables as well as integrating data from design and qualification tests.

NOMENCLATURE

ASS = AutoTrak steering system
BCPM = Bi-directional communication and power module
BHA = Bottomhole assembly
HALT = Highly accelerated life test
HAST = Highly accelerated stress test
IRMLE= Iteratively reweighted maximum likelihood estimation.
LVPS = Low voltage power supply
LWD = Logging while drilling
MaPS = Maintenance and performance system
MLE = Maximum likelihood estimation
MWD = Measurement while drilling
PCBA = Printed circuit board assembly
PHM = Prognostics and health management
PoF = Physics of failure
RPM = Revolutions per minute
F = Failure
L = Lateral vibration
$M_i = i^{th}$ model identifier
N = Symbol used to represent negative decision, generally "no" or "0"
S = Symbol used to represent stick slip or suspensions
T = Temperature
X = Vector of parameters like temperature and vibrations
Y = Symbol used to represent affirmative decision, generally "yes" or "1"
f = Probability density function
m = Number of models
n = Number of records
p = Probability
$p(a|b)$ = Conditional probability of occurrence of event a provided b is true
revid = Revision identifier
tf = Time to failure (drilling hours)
w_i = Weight of i^{th} data point

x_{ave} = Average value of parameter x
x_{stdev} = Standard deviation of parameter x
α = Calibration parameters of reliability model
$\mathcal{L}$ = Likelihood
η = Characteristic life or scale factor of a probability distribution
β = Shape factor of a probability distribution
σ = Standard deviation
λ= Hazard function
{CF} = Set of life data for confirmed failure
{O} = Set of outliers
{S} = Set of life data for suspension
{UF} = Set of life data for unconfirmed failure
Load, *Stress* and *Severity* are used interchangeably to describe the impact of an operational environment (mechanical and thermal) on the durability of parts.
Nominal part is a representative part that has a life equal to the average of several parts produced using the same manufacturing process and operating under the same condition.
Run refers to a drilling mission that can last for several hours.
Suspensions are used in reliability modeling to represent hours accumulated on parts that are in operation or removed from service for reasons other than failure.

REFERENCES

Bailey, C., Tilford, T., Lu, H., (2007), Reliability analysis for power electronics modules. *IEEE 30th International Spring Seminar on Electronics Technology*. 9-13 May 2007, Cluj-Napoca, doi: 10.1109/ISSE.2007.4432809.

Baker Hughes Incorporated. (2010), *Repair and Maintenance Return Policy for Printed Circuit Board Assemblies*. Document RM-002, Houston TX, USA.

Baker Hughes Incorporated (2008), *OnTrak Repair & Maintenance Manual*, Document *OTK-10-0500-001*, Houston TX, USA.

Barker, D., Dasgupta, A., Pecht, M., (1992), PWB solder joint life calculations under thermal and vibrational loading, *Journal of The IES*, Vol. 35, No.1, February 1992, pp. 17-25. Doi: 10.1109/ARMS.1991.154479.

Born, F., and Boenning, R., A., (1989), Marginal checking – A technique to detect incipient failures, *Proceedings of the IEEE Aerospace and Electronics Conference*, 22-26 May 1989, pp. 1880 – 1886. Doi. 10.1109/NAECON.1989.40473

Chatterjee, K., Modarres, M., Bernstein, J., B., (2012), Fifty years of physics of failure, *Journal of Reliability Information Analysis Center*, Vol: 20 #1. Doi: 10.1109/RAMS.2013.6517624.

Dasgupta, A., (1993), Failure mechanism models for cyclic fatigue, *IEEE Transactions on Reliability*, Vol. 42, No. 4, December 1993, pp. 548-555. Doi: 10.1109/24.273577.

Duffek D., (2004), *Effect of Combined Thermal and Mechanical Loading on the Fatigue of Solder Joints*. Master's Thesis. University of Notre Dame, IN, USA.

Evans, J., Lall, P., Bauernschub, R., (1995), A framework for reliability modeling of electronics. *Proceedings of IEEE Annual Reliability and Maintainability Symposium*, January 1995, Washington D. C., USA. doi 10.1109/RAMS.1995.513238.

Garvey, D., R., Baumann, J., Lehr, J., Hines, J., W., (2009), Pattern recognition based remaining useful life estimation of bottom hole assembly tools. *SPE/IADC Drilling Conference and Exhibition*, 2009, Amsterdam, The Netherlands. Doi: 10.1109/24.273577.

Gingerich, B., L., Brusius, P., G., Maclean, I., M., (1999), Reliable electronics for high-temperature downhole applications. *SPE Annual Technical Conference and Exhibition*, 1999, Houston, Texas.

Hu, J., M., Pecht, M., Dasgupta, A., (1991), A probabilistic approach for predicting thermal fatigue life of wire bonding in microelectronics, *ASME Journal of Electronics Packaging*, Vol. 113, 1991, pp. 275-285. doi:10.1115/1.2905407.

Kalgren, P., W., Baybutt, M., Ginart, A. (2007), Application of prognostic health management in digital electronic systems. *IEEE Aerospace Conference*, Big Sky, Montana. Doi 10.1109/AERO.2007.352883.

Lall, P., Singh, N., Strickland, M., Blanche, J., Suhling, J., (2005), Decision-support models for thermo-mechanical reliability of lead-free flip-chip electronics in extreme environment. *Proceedings of 55th Electronics Components and Technology Conference*, Lake Buena Vista, FL, USA. Doi: 10.1109/ECTC.2005.1441257.

Lall, P. (1996), Temperature as an input to microelectronics reliability models. *IEEE Transactions on Reliability*, vol. 45, no. 1, pp. 3-9.

Lall, P., Choudhary, P., Gupte, S., Suhling, J., Hofmeister, J. (2007), Statistical pattern recognition and built-in reliability test for feature extraction and health monitoring of electronics under shock loads. *Proceedings of 57th IEEE, Electronic Components and Technology Conference*, 2007, Sparks, Nevada. Doi: 10.1109/ECTC.2007.373942

Mirgkizoudi, M., Changqing, L., Riches, S., (2010), Reliability testing of electronic packages in harsh environments. *Proceedings of 12th Electronics Packaging Technology Conference*, 2010. Doi: 10.1109/EPTC.2010.5702637

Mishra, S. and Pecht, M. (2002), In-situ sensors for product reliability monitoring, *Proceedings of SPIE*, Vol. 4755, 2002, pp. 10-19. Doi: 10.1117/12.462807

Nasser, L., Curtin, M. (2006), Electronics reliability prognosis through material modeling and simulation, *IEEE Aerospace Conference*, Big Sky,

Montana. Doi: 10.1109/AERO.2006.1656125

Normann, R. A., Henfling, J. A., Chavira, D. J. (2005), Recent advancements in high-temperature, high-reliability electronics will alter geothermal exploration. *Proceedings World Geothermal Congress*, Antalya, Turkey.

Osterman, M. (2001), We still have a headache with arrhenius, *Electronics Cooling*, Vol. 7, Number 1, pp. 53-54, February 2001.

Pecht, M., Radojcic, R., Rao, G. (1999), *Guidebook for managing silicon chip reliability*, CRC Press, Boca Raton, FL.

Pecht, M., Lall, P., Hakim, E. (1997), *Influence of temperature on microelectronics and system reliability*, CRC Press, New York, NY

Ridgetop Semiconductor-Sentinel Silicon ™ Library, "Hot Carrier (HC) Prognostic Cell," August 2004

Shinohara, K., Yu, Q. (2010), Evaluation of fatigue life of semiconductor power device by power cycle test and thermal cycle test using finite element analysis. Engineering, 2010, 2, 1006-1018. Doi: 10.4236/eng.2010.212127.

Sutherland, H., Repoff, T., House, M., and Flickinger, G., Prognostics, a new look at statistical life prediction for condition-based maintenance, *IEEE Aerospace Conference*, 2003. Volume: 7-3131, March 8-15, 2003. Doi: 10.1109/AERO.2003.1234156.

Vichare, N. M. (2006), *Prognosis and Health Management of Electronics by Utilizing Environmental and Usage Loads*, Doctoral dissertation. 2006, University of Maryland, College Park.

Vichare, N., Rodgers, P., Eveloy, V., Pecht, M., Environment and Usage Monitoring of Electronic Products for Health Assessment and Product Design, *Journal of Quality Technology and Quality Management*, Vol. 4, No. 2, pp. 235-250, 2007.

Vijayaragavan, N. (2003), *Physics of Failure Based Reliability Assessment of Printed Circuit Boards used in Permanent Downhole Monitoring Sensor Gauges*. Master dissertation. University of Maryland, College Park, USA.

Wassell, M., Stroehlein, B. (2010), Method of establishing vibration limits and determining accumulative vibration damage in drilling tools. *SPE Annual Technical Conference and Exhibition*, September 2010, Florence, Italy. Doi: 10.2118/135410-MS

White, M., Bernstein, J. B. (2008), Microelectronics reliability: Physics-of-failure based modeling and lifetime evaluation. *NASA Joint Propulsion Laboratory Report*, Project Number: 102197.

Wong, K. L. (1995), A new framework for part failure rate prediction models. *IEEE Transactions on Reliability*, 44(1):139-145, March. Doi: 10.1109/24.376540

Young, D., Christou, A. (1994), Failure mechanism models for electromigration, *IEEE Transactions on Reliability*, Vol. 43, No. 2, pp. 186 – 192. Doi 10.1109/24.294986

Zhang, H., Kang, R., Pecht, M. (2009), A hybrid prognostics and health management approach for condition based maintenance. *IEEE International Conference on Industrial Engineering and Engineering Management*, pp1165–1169. Doi 10.1109/IEEM.2009.5372976.

Zhang R., Mahadevan S., 2000, Model uncertainty and bayesian updating in reliability–based inspection. Structural Safety 22, 145-160.doi 10.1016/S0167-4730(00)00005-9.

BIOGRAPHIES

Amit A. Kale was born in Bhopal, India on October 25 1978. He earned PhD in 2005 and MS in 2004 in Mechanical Engineering from University of Florida, Gainesville, Florida, USA and BTech in Aerospace engineering from Indian Institute of Technology, Kharagpur, India in 2000. He joined Baker Hughes Inc. in 2012 and currently works on health prognostics of drilling system in Houston, Texas. Prior to that he worked in GE Global Research, Niskayuna, New York from 2005-2012.

Katrina Carter-Journet was born in Baton Rouge, Louisiana. She has a BS in Physics from Southern University in Baton Rouge, Louisiana (USA) and a MS in Biophysics from Cornell University in Ithaca, New York (USA). Her work experience has been in the biomedical engineering, aerospace, and the oil and gas industries. Currently, she works on developing and maintaining life prediction methodologies to improve the maintenance process and retirement of tools used to support drilling and evaluation services.

Troy Falgout was born on 10 December 1967 in Erath, Louisiana. He holds an Associate's Degree in Electronics form Southern Technical College Lafayette, La 1987 and Bachelor Degree in Business Management from University of Phoenix 2014. He has been working with Baker Hughes since 1989 as a Technician, Tech Support Engineer, Maintenance Manager and Reliability Manager for Drilling Services.

Ludger E. Heuermann-Kühn was born in Twistringen, Germany on November 18th 1968. He earned a BSc in Mechanical Engineering from the University in Sunderland, UK and Diplom Ingenieur (FH) from the Fachhochschule Kiel, Germany. He joined Baker Hughes in 1997 and is currently the manager of Central Reliability Assurance division for drilling service. Prior to that he worked in different engineering and managerial positions in technical services, product development and product reliability engineering.

Derick Zurcher is the Product Line Manager for Baker Hughes Logging While Drilling Formation Evaluation

services. He has 17 years industry experience, with prior roles in Geoscience and LWD Operations. He has a BSc in Geology from the University of South Australia, an MSc in Petroleum Geology from NCPGG, and an MBA from London Business School. He is a member of the SPWLA and SPE Century Club.

APPENDIX A

A. General Log-Linear Model

The relation between characteristics of life and stress variables are represented by using one of the three models: generalized log-linear (*GLL*), proportional hazard (*PH*) or cumulative damage (*CD*). The *GLL* model represents life using Eq. (A-1)

$$\eta(\bar{x}) = e^{a_0+\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} a_{i,j}\, x_i x_j} \quad \text{(A-1)}$$

Where $\bar{x} = \{T, L, S\}$. For a Weibull distribution, the probability density function is shown in Eq. (A-2), where β is the shape parameter, η is the scale parameter and α*'s* are unknown parameters calculated from field data using the maximum likelihood estimation technique.

$$f(t,\bar{x}) = \beta t^{\beta-1} e^{-\beta.\eta(\bar{x})} e^{-t^{\beta} e^{-\beta.\eta(\bar{x})}} \quad \text{(A-2)}$$

The probability density function (PDF) for an exponential distribution can be obtained by putting $\beta=1$ in Eq. (A-1). For lognormal distribution, the probability density function for a *GLL* stress function is shown in Eq. (A-3):

$$f(t,\bar{x}) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{ln(t)-\eta(\bar{x})}{\sigma}\right)^2} \quad \text{(A-3)}$$

B. Proportional Hazard Model

For a proportional hazard model, the hazard rate of a component is affected by hours in operation and stress variables. The instantaneous hazard rate of a part is given by the equation as:

$$\lambda(t,\bar{x}) = \frac{f(t,\bar{x})}{R(t,\bar{x})} = \lambda_0(t)\,\eta(\bar{x},\bar{a}) \quad \text{(A-4)}$$

where f is the probability density function and R is the reliability function. The instantaneous hazard rate λ_0 is a function of time only and the stress function η is function of operating stresses like temperature or vibration. The list of unknown model parameter $\bar{a}$ is obtained by calibrating model-to-test data using maximum likelihood estimation (MLE). The stress function η is given by Eq. (A-5):

$$\eta(\bar{x}) = e^{\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} a_{i,j}\, x_i x_j} \quad \text{(A-5)}$$

Substituting Eq. (A-5) in Eq. (A-2), the hazard function can be written for a Weibull distribution using Eq. (A-6):

$$\lambda(t,\bar{x}) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} a_{i,j}\, x_i x_j} \quad \text{(A-6)}$$

C. Cumulative Damage Model

The cumulative damage model is designed to incorporate the effect of varying stress on life of components. The model takes into account the impact of damage accumulated at each stress level on the reliability of the part. Damage accumulation can take place at different rates for different stress levels and can be determined using the linear damage sum (Miner's rule), inverse power law or cycle counting techniques like rain flow counting. The cumulative damage model used in the paper is established from Miner's rule, which is based on the hypothesis that if there are n different stress levels and the time to failure at the i^{th} stress σ_i is Tf_i, then the damage fraction, p, is given by Eq. (A-7):

$$p = \sum_{i=1}^{n} \frac{t_i}{Tf_i} \quad \text{(A-7)}$$

Where t_i is the number of cycles accumulated at stress σ_i and failure occurs when the damage fraction equals unity. The probability distribution functions for Weibull and lognormal distributions are obtained by substituting Eq. (A-7) in Eqs (A-2) and (A-3), respectively. Given the stress variables $\bar{\boldsymbol{X}} = \{T, L, S, RPM, L\times T, S\times T, L\times S, S\times RPM\}$, the PDF for a Weibull distribution is given by:

$$I(t,\bar{x}) = \int_0^t \frac{e^{a_0+\sum_{i=1}^{n} a_i x_i(t) + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} a_{i,j}\, x_i(t) x_j(t)}}{p} dt$$

$$f(t,\bar{x}) = \beta s(t,\bar{x})\big(I(t,\bar{x})\big)^{\beta-1} e^{-\left((I(t,\bar{x}))\right)^{\beta}} \quad \text{(A-8)}$$

D. Characteristic Life Function

The life characteristic function describes a general relation between failure time and stress levels. The life characteristic can be any time-to-failure measure such as the mean, median or hazard rate that represents a bulk property of a probability distribution. Ideally, the function incorporates the governing equations that represent the physical phenomenon of degradation of the material under application of load. Typical electronic circuit boards used in drilling and evaluations are complex and the governing equations representing degradation and failure mechanisms are difficult to model; hence, the paper evaluates several empirical functions between stress variables and selects the one that best fits the field data.

E. Maximum Likelihood Estimation and Outlier Detection

The maximum likelihood estimation (MLE) obtains the most likely values of parameters that best describes lifecycle data. Typically, the life cycle data of a part contain two sets of populations (a) hours to failure on samples that failed in

an experiment or in field and (b) hours in operation for parts that are either currently being operated or those that are retired for precautionary measures but were fully functional at that time.

$$ln(L) = \sum_{i=1}^{F_e} W_i^F \times N_i^F \times ln\big(f(T_i,\eta,\beta)\big) + \sum_{i=1}^{S} W_i^S \times N_i^S \times ln\big(1 - F(Ts_i,\eta,\beta)\big) - \sum_{i=1}^{I} W_i^I \times N_i^I \times ln\Big\{\big(1 - F(T_i^L,\eta,\beta)\big) - \big(1 - F(T_i^R,\eta,\beta)\big)\Big\} \quad \text{(A-9)}$$

Where the initial weight of each data point is given by

$$W_i^* = \frac{1}{\sum_{i=1}^{F_e} N_i^F + \sum_{i=1}^{S} N_i^S + \sum_{i=1}^{I} N_i^I} \quad \text{(A-10)}$$

F_e is the number of samples for which the exact times-to-failure is known, N_i^F is the number samples for which the exact time-to-failure is T_i, f is the probability density function (pdf) for time to failure, η is the scale factor and β shape factor of the pdf, N_i^S is the number samples for which the right censoring time is Ts_i, N_i^I is the number samples for which the left censoring time is T_i^L and right censoring time is T_i^R . The W_i^* is the weight of i^{th} data subgroup is determined by the IRMLE algorithm. The outliers identified by the algorithm are shown in Fig. A1-Fig. A6 and the comparison of estimated life versus actual drilling hours to failure is shown in Fig. A7.

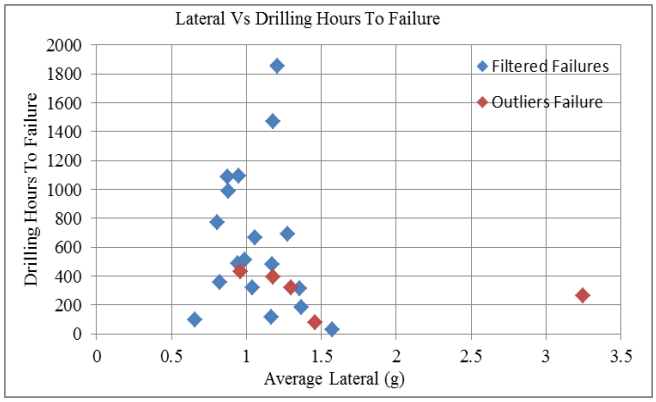


Figure A1. Time to failure Vs. lateral vibration severity for fielded LVPS-modem serialized parts.

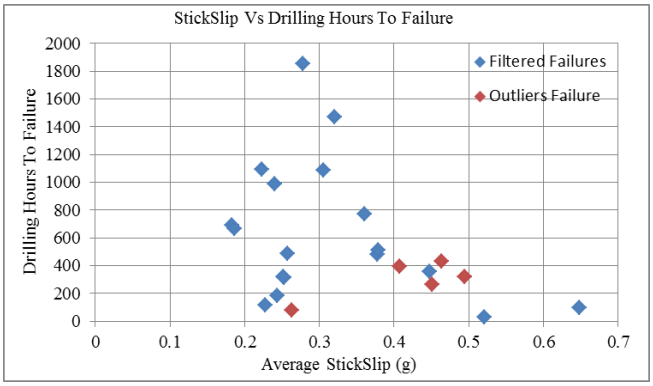


Figure A2. Time to failure Vs. stickslip vibration severity for fielded LVPS-modem serialized parts.

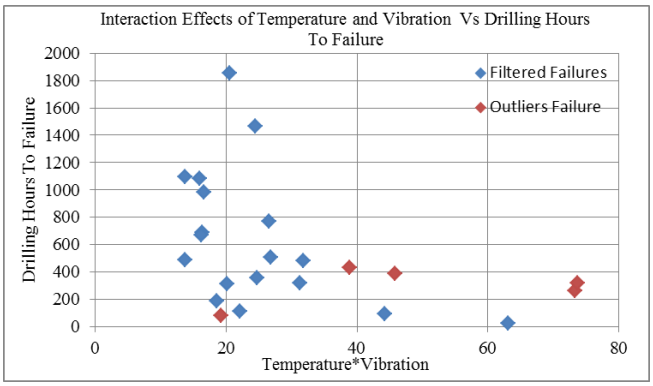


Figure A3. Impact of interaction of temperature and vibration on failure of LVPS-modem serialized parts.

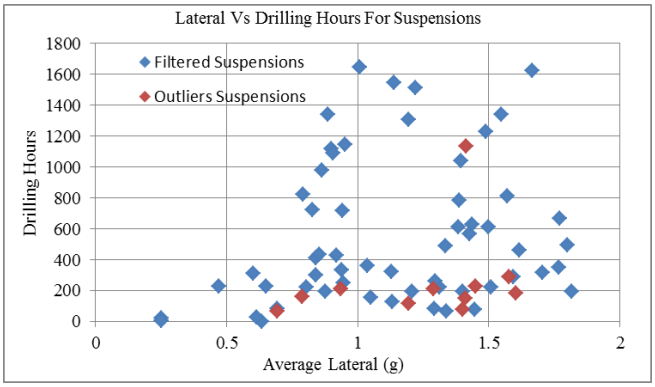


Figure A4. Suspension time Vs. lateral vibration severity for fielded LVPS-modem serialized parts.

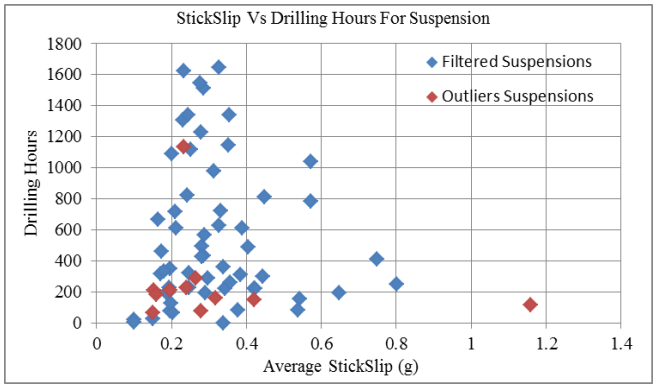


Figure A5. Suspension time Vs. stickslip vibration severity for fielded LVPS-modem serialized parts.

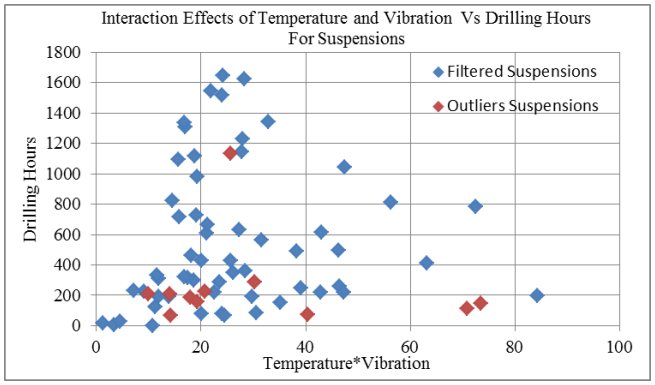


Figure A6. Suspension time Vs. interaction effect for fielded LVPS-modem serialized parts.

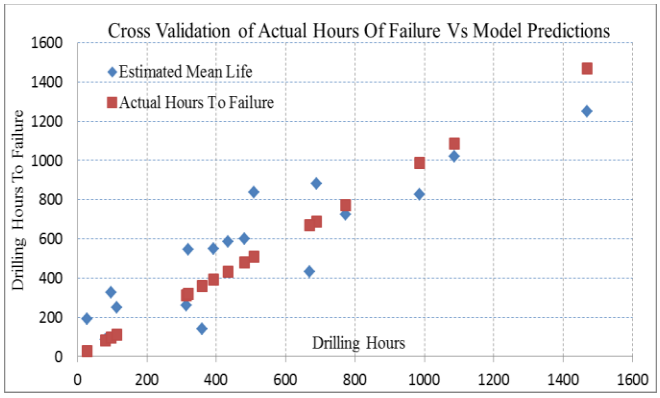


Figure A7. Comparison of actual life Vs. predicted mean life for parts that failed in field

Are Current Prognostic Performance Evaluation Practices Sufficient and Meaningful?

Shankar Sankararaman[1], Abhinav Saxena[2], and Kai Goebel[3]

[1,2] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov
abhinav.saxena@nasa.gov

[3] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

This paper investigates the shortcomings of performance evaluation for prognostic algorithms, particularly in the presence of uncertainty. To that end, the various elements of a prognostic algorithm (present health state estimation, future load condition, degradation model, and damage threshold) and their effects on prognostics are examined. Each of these elements contribute to overall prediction performance and therefore it is important to distinguish between (1) assessment of the correctness of information regarding these quantities, and (2) the assessment of correctness of the prognostic algorithm. The need for proper accounting for uncertainty in the various associated elements is discussed. Next, the shortcomings of traditional comparisons between ground truth and algorithm prediction is discussed. Several scenarios are pointed out where misleading interpretations about evaluation outcomes are possible. In order to address these shortcomings an "informed evaluation" methodology is being proposed, where the algorithm is informed with future loading/operating conditions before comparing against ground truth. Additionally, the importance of estimating the accuracy of aggregating the different sources of uncertainty using rigorous mathematical procedures is also emphasized. While this discussion does not target developing new metrics, it highlights key criteria for an accurate performance evaluation process under uncertainty and proposes new measures to accomplish this goal.

1. INTRODUCTION

1.1. Prognostics

Prognostics, the ability to predict future events, conditional on anticipated usage and environmental conditions, significantly contributes to a system's resilience for safe and efficient operation. It is now well accepted that prognostics can add considerable value to life cycle cost reduction by assessing the state of health of the system components, and estimating their remaining useful life that makes it possible to initiate a mitigating action that will either prevent the breakdown, minimize downtime, avoid unscheduled maintenance, or result in similar outcomes that minimize operational cost of the system. However, at the same time, prognostics is inherently affected by various sources of uncertainty present in the system; if the methods that deal with uncertainty are not adequately understood and incorporated, it can be difficult to make reliable predictions with high accuracy and confidence. It is, therefore, not surprising that considerable attention has been given to this technology in the last few years. A variety of different approaches have been explored and employed to predict system health and/or estimate remaining useful life. However, it is important to note that the term "prognostics" has been used by various practitioners in any context that has a predictive element but not all of these methods result in estimation of remaining life. Subsequently, it also has a bearing on the interpretation and treatment of uncertainty in each of these methods, which is important not only to understand how to incorporate these uncertainties in the analysis but also to assess performance of these methods in a technically correct and rigorous manner (Saxena, Sankararaman, & Goebel, 2014).

1.2. Prognostic Performance Evaluation

Performance assessment of prognostics algorithm is an indispensable element in maturing prognostics and health management technology as these predictions become the basis of any subsequent decision making process. Mitigating actions taken based on these decisions ultimately determine the effectiveness of the overall health management system. Most of the existing literature on prognostics performance evaluation

Shankar Sankararaman et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

focuses on choosing the most appropriate metrics to evaluate algorithms. Several metrics have been proposed and used in the past that measure unique characteristics of prognostics (Saxena, Celaya, Saha, Saha, & Goebel, 2010). These metrics described different ways to express and measure accuracy, precision, timeliness, and prediction-confidence attributes of the prediction of a prognostic algorithm. Less attention has been paid towards determining the correct approach for evaluating and interpreting prognostic performance under uncertainty. Current approaches rely on comparing predicted outcomes to observed end of life (also referred to as ground truth). The key question, as investigated in this paper, is whether such a comparison is technically correct, especially when considering uncertainty in the prediction process. In contrast to discussing prognostic metrics, this paper attempts to identify a meaningful approach for performance evaluation irrespective of which metrics are used to quantify performance. In particular, two issues are explored: (1) choosing the baseline to compare prediction results with and (2) identifying a method that can be used to obtain such information. In the process, several important caveats in interpreting the results of prognostic algorithms are explained in detail and several misconceptions are clarified in this regard.

1.3. Relation to Work on Metrics

For providing a clear context with regards to earlier works investigating prognostic performance, it is important to draw connections between the *what should be measured* and *how prognostic metrics were designed*. Early versions of prognostics algorithms output were point estimates of end-of-life that were compared with the observed end-of-life to assess performance (Saxena et al., 2008). Later as prognostics algorithms matured they started incorporating uncertainties in predictions through various representations of uncertainty, although mostly dominated by probability distributions. However, the basic underlying question of what the key contributing factors to the quality of a prediction are and how the contribution of each can be evaluated separately have not been addressed in detail until very recently (Sankararaman & Goebel, 2013b). Prognostic performance is understood to depend on two distinct factors; 1) External inputs (data quality, operating environment, system loading, etc.), and 2) Internal processing (fault models, state estimation methods, uncertainty propagation methods, etc.). To gain full understanding of uncertainty expressed in remaining useful life (RUL) estimates it is important to isolate the effects of these different internal and external factors through adequate performance evaluation while algorithm development. Based on feedback from such evaluation, targets for further technology improvement can be identified and a baseline of acceptable performance can be established before a prognostic system is put into usage. This paper extends the discussion in (Saxena et al., 2014) by focusing on effects of uncertainty in prognostics for the purpose of performance evaluation and explores how carefully designed performance evaluation process can help distill these effects.

1.4. Organization of this Paper

This paper focuses its attention on performance evaluation of only condition based prediction methods for prognostics. Other prediction methods are considered beyond the scope of this paper. First, Section 2 describes various sources of uncertainty that are present in prognostics and clearly distinguishes between the interpretation of uncertainty in condition-based prognostics and fleet-based prediction methods. This discussion dissects the overall uncertainty into a few fundamental elements and subsequently provides a stepwise approach to assess prognostic performance so that these effects of each of these elements on prognostic performance evaluation can be assessed. Next, Section 3 discusses the impact of uncertainty on prognostic algorithms through an illustrative example and a simple prediction algorithm. Section 4 explains the challenges involved in performance evaluation of prognostic algorithms and Section 5 explains different types of performance measures. Section 6 numerically illustrates the above concepts using a lithium-ion battery application. Finally, conclusion and future work are presented in Section 7.

2. PROGNOSTIC ALGORITHMS

In order to completely understand the various aspects of performance evaluation of prognostic algorithms, it is necessary to understand the various elements of a prognostic algorithm. A prognostic algorithm ideally takes all available information (state estimate, future estimates, degradation model, etc.) and computes the remaining useful life of the component or system of interest.

2.1. Key Elements of a Prognostic Algorithm

For the purpose of rating the performance of an algorithm, it is important to decide which elements are part of an algorithm and which are not. Roychoudhury et al. (Roychoudhury, Saxena, Celaya, & Goebel, 2013) focused on identifying the key aspects of a prognostic algorithm, this argument is extended in this paper to identify the various elements that are needed to determine the remaining useful life, as follows:

1. Present condition (state) of the system/component
2. Future (operational, loading, environmental, etc.) conditions of the system/component
3. Degradation model of the system/component
4. End-of-Life damage threshold
5. The actual algorithmic procedure, that combines the above information systematically in order to compute the remaining useful life.

One could argue that quantifying the present condition of the system/component through a state estimation algorithm (perhaps using a Bayesian filtering approach such as particle filtering or Kalman filtering) is a necessary and essential component of the prognostic algorithm. However, the development of the degradation model and estimating the future conditions seem to be outside the scope of the prognostic algorithm. The problem is that these two components are "inputs" to a prognostic algorithm, i.e., the algorithm needs these two pieces of information to predict the remaining useful life. It would not reasonable to penalize an algorithm whose predictions do not compare well with ground truth data, if the algorithm did not have access to an accurate degradation model and/or an accurate estimate of the future conditions of the component/system. Similarly, it is not reasonable to accept a prognostic algorithm whose predictions apparently match well with ground truth data, if the algorithm had used inaccurate future conditions and inaccurate degradation model (whose inaccuracies could cancel each other out). For example, the degradation model may have a much smaller degradation rate and the chosen future conditions may be much more severe than reality.

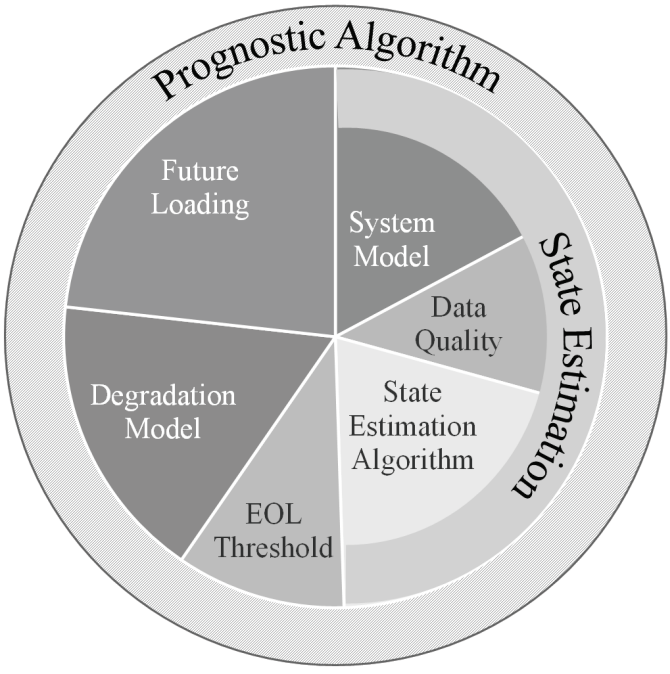


Figure 1. Components of Prognostics Algorithm

Therefore, this paper explores the various aspects of performance evaluation with an emphasis on the above elements of a typical prognostic algorithm, as explained through the rest of this paper.

2.2. Uncertainty in Prognostics

While non-probabilistic methods (Wang, 2011) such as Fuzzy logic, possibility theory, Dempster-Shafer theory, Evidence theory, etc. have been used for the treatment of uncertainty, probabilistic methods have been predominantly used for uncertainty representation in prognostics (DeCastro, 2009; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Saha, Goebel, Poll, & Christophersen, 2009). Without loss of generality, the rest of this paper will focus only on prognostic algorithms based on probability theory.

In order to evaluate the performance of prognostic algorithms in the presence of uncertainty, it is important to answer questions such as:

1. What does one actually mean by "uncertainty" in prognostics?
2. What causes uncertainty in prognostics?
3. What are various elements of a prognostic algorithm that are affected by uncertainty?
4. What is the contribution of these elements to overall prognostic performance?

2.3. Interpreting Uncertainty in Prognostics

Though mathematical axioms and theorems of probability have been well-established in the literature and probabilistic methods are being increasingly used for uncertainty quantification in engineering, there is considerable disagreement among researchers on the interpretation of probability. There are two major interpretations based on physical and subjective probabilities, respectively. Physical probabilities (Szabó, 2007), also referred to objective or frequentist probabilities, are related to random physical systems such as rolling dice, tossing coins, roulette wheels, etc. Each trial of the experiment leads to an event (which is a subset of the sample space), and in the long run of repeated trials, each event tends to occur at a persistent rate, and this rate is referred to as the relative frequency". These relative frequencies are expressed and explained in terms of physical probabilities. Thus, physical probabilities are defined only in the context of random experiments. On the other hand, subjective probabilities (De Finetti & de Finetti, 1977) can be assigned to any "statement". It is not necessary that the concerned statement is in regard to an event which is a possible outcome of a random experiment. In fact, subjective probabilities can be assigned even in the absence of random experiments. The Bayesian methodology is based on subjective probabilities, which are simply considered to be degrees of belief and quantify the extent to which the statement is supported by existing knowledge and available evidence. Calvetti and Somersalo (Calvetti & Somersalo, 2007) explain that "randomness" in the context of physical probabilities is equivalent to "lack of information" in the context of subjective probabilities. In this approach, even deterministic quantities can be represented using probability

distributions which reflect the subjective degree of the analyst's belief regarding such quantities.

This leads to the obvious question - is one particular interpretation more suitable to prognostics? In general, both interpretations may be suitable. However, in the particular context of condition-based monitoring or online health monitoring, there is only one system which is being monitored, and hence, at any time instant, there is no "physical randomness" associated with the system (from a frequentist point of view). Therefore, any quantity associated with a system, even though it may be uncertain, cannot be represented using a probability distribution, following the frequentist interpretation of probability. Nevertheless, system state estimation during health monitoring is commonly performed using particle filters and Kalman filters, and these approaches compute probability distributions for the state variables; therefore, the only possible explanation for such calculation is that the subjective (Bayesian) approach is being inherently used for uncertainty quantification. Such filtering approaches are known as "Bayesian tracking" methods not only because they make use of Bayes theorem, but also fall within the realm of subjective probability. This implies that the uncertainty estimated through the aforementioned filtering algorithms are simply reflective of the analyst's degree of belief, and not related to actual physical probabilities. Similarly, the uncertainty in future conditions (loading, operating, and environmental conditions) also need to interpreted subjectively. For example, if the anticipated current on a battery follows a normal distribution with mean and standard deviation equal to 10 and 1 (current units) respectively, then this probability distribution is only reflective of the subjective belief, and only one realization may occur in reality. The actual current may be 10 units (which is not possible to know), and this implies that the subjective belief was reasonable; the subjective belief would have been even better had the standard deviation been smaller. On the other hand if the actual current had been 30 units, then it implies that the subjective belief was completely wrong.

Sometimes, in practice, both frequentist and subjective information can be useful, even in condition-based prognostics. For example, an ensemble of test units may be used to develop degradation models and learn the corresponding model parameters. Since these models and their parameters are estimated based on physically variable units, the uncertainty in such parameters need to be interpreted from a frequentist point of view. However, when such a model is used in condition-based monitoring, these parameters are typically updated in order to reflect the parameters of the particular unit; during this procedure, the interpretation of uncertainty transitions from "frequentist" to "subjective" as the information described in terms of uncertainty changes from reflecting the ensemble of test units to the particular unit under consideration for condition-based monitoring. It is important to understand the interpretation of uncertainty during the course of the monitoring procedure, depending upon what information is used to characterize and quantify the aforementioned uncertainty.

2.4. Sources of Uncertainty in Prognostics

Having discussed the importance and interpretation of uncertainty, this subsection seeks the answer to the question: What are the different sources of uncertainty in prognostics? Typically, the answer to this question varies from application to application, and depends on the type of prediction. For example, in testing-based prediction methods (referred to as "reliability-based testing" in some publications), the remaining useful life is typically calculated by testing multiple nominally identical specimens of the engineering component/system. It may be noted that the term "remaining" in "remaining useful life" may not be applicable to such testing methods. This is because, testing is typically carried out before the engineering system is under operation. The term "time-to-failure" is more appropriate for testing-based health management. It is important not to confound "time-to-failure" and "remaining useful life".

Assume that a set of run to failure experiments have been performed with high level of control, ensuring same usage and operating conditions. The time to failure for all the n samples ($r_i; i = 1$ to n) are measured. It is important to understand that *different* time-to-failure values are obtained due to inherent variability across the n different specimens, thereby confirming the presence of physical probabilities or true randomness. The various factors that contribute are:

1. Inherent variability in properties and characteristics of the nominally identical specimens
2. Inherent variability across the loading conditions experienced by each of the individual specimens
3. Inherent variability in operating and environmental conditions for each of the individual specimens

On the other hand, in condition-based prognostics, the focus should be on monitoring the performance of one particular component/system where the inherent variability across nominally identical units are not of interest. In other words the end of life of the system under test is not governed by system to system variability within the context of condition based predictions or prognostics. It is, therefore, necessary to adopt a significantly different approach for the treatment of uncertainty. Various uncertainties involved in prognostics can be divided into following broad categories:

1. **Present uncertainty:** Prior to prognosis, it is important to be able to precisely estimate the condition/state of the component/system at the time at which RUL needs to be predicted. Typically, damage (or faults) are expressed in terms of states, and therefore, estimating the state is equivalent to estimating the extent of damage (or fault).

This is related to state estimation and is commonly addressed using filtering. Output data (usually collected through sensors) are used to estimate the state and many filtering approaches (Kalman filtering, particle filtering, etc.) are able to provide an estimate of the uncertainty in the state. Practically, it is possible to improve the estimate of the states and thereby reduce this uncertainty, by using better sensors and improved filtering approaches. It is important to understand that the system is at a particular state at any time instant, and the aforementioned uncertainty simply describes the lack of knowledge regarding the "true" state of the system.

2. **Future uncertainty:** The most important source of uncertainty in the context of prognostics is due to the fact that the future is unknown, i.e. the loading, operating, environmental, and usage conditions are not known precisely, and it is important to assess this uncertainty before performing prognosis. If there is no uncertainty regarding the future, then there would be no uncertainty regarding the *true* remaining useful life of the engineering component/system. However, this true RUL needs to be estimated using a model; the usage of a model imparts additional uncertainty as explained below.
3. **Modeling uncertainty:** It is necessary to use a functional degradation model in order to predict future state behavior, i.e., model the response of the system to anticipated loading, environmental, operational, and usage conditions. Further, the end-of-life is also defined using a Boolean threshold functional model, that is used to indicate whether failure has occurred or not. These two models are jointly used to predict the RUL, and they may either be physics-based or data-driven. It may be practically impossible to develop models that accurately predict the underlying reality. Modeling uncertainty represents the difference between the predicted response and the true response (that can neither be known nor measured accurately), and comprises of several parts: model form, model parameters, and process noise. While it may be possible to quantify these terms until the time of prediction, it is challenging to know their values at future time instants.

3. IMPACT OF UNCERTAINTY ON PROGNOSTIC ALGORITHMS

To better illustrate the impact of uncertainty on prognostic algorithms, a conceptual example is introduced in this section.

3.1. Conceptual Example

Consider an engineering component whose health state at any time instant is given by $x(t)$. Consider a simple degradation model, where the rate of degradation of the health state (that decreases with time, due to the presence of damage) is proportional to the current health state. This can be mathematically expressed as:

$$\dot{x}(t) \propto x(t), \tag{1}$$

where the constant of proportionality is a negative number. Since differential equations are usually solved by considering discrete time instants, the above equation can be rewritten as:

$$x(k+1) = a.x(k) + b, \tag{2}$$

where k represents the discretized time-index. The condition that "the constant of proportionality in Eq. 1 is negative" is equivalent to the condition that "$a < 1$ in Eq. 2". The initial health state, i.e., $x(0)$ is a random variable, and is expressed using a probability distribution. For the sake of illustration, let a denote the loading on the system (the smaller the value a, the larger the degradation rate), and let b denote the parameter of the above degradation model. While a and b are constant and time-invariant (for the sake of illustrating the conceptual example), they are random and expressed using probability distributions. (In practical examples, the probability distributions of a and b could vary as a function of time.)

In order to compute the remaining useful life, it is necessary to chose a threshold function that defines the occurrence of failure. Since $x(k)$ is a decreasing function, the threshold function will indicate that failure occurs when the state value x becomes smaller than a critical lower bound (l), and the first time instant at which this event occurs indicates the end of life, and this time instant can be used to calculate the RUL. For the purpose of illustration, consider prediction at the initial time instant; hence, the end of life is equal to the remaining useful life. This remaining useful life (r, an instance of the random variable R) is equal to the smallest n such that $x(n) < l$, and is expressed as:

$$r = \inf\{n : x(n) < l\}, \tag{3}$$

In general (i.e., at arbitrary time instants when it is desired to make prediction), the RUL is calculated as the difference between the end-of-life and the time of prediction.

3.2. Closed-Form Solutions?

This section postulates that closed-form analytical solutions for the remaining useful life prediction are not available even for such simple problems involving linear prediction models. In order to illustrate this point, assume that the chosen time-discretization level is infinitesimally small, it is possible to directly estimate the RUL by solving the equation:

$$a^r.x(0) + \sum_{j=0}^{j=r-1} a^j.b = l. \tag{4}$$

The above equation can be used to calculate the RUL (r) as a function of the initial state ($x(0)$), loading (a) and model parameter (b). For the sake of further simplification, assume

that a and b are completely known constants and $x(0)$ is the only uncertain quantity; further assume that $x(0)$ follows a Gaussian distribution. The following analysis shows that it is impossible to analytically calculate the remaining useful life prediction even with only one uncertain variable and a linear degradation model.

The RUL R follows a Gaussian distribution if and only if it is linearly dependent on $x(0)$. In other words, R follows a Gaussian distribution if and only if Eq. 4 can be rewritten as:

$$\alpha.r + \beta.x(0) + \gamma = 0 \tag{5}$$

for some arbitrary values of α, β, and γ. If it were possible to estimate such values for α, β, and γ, the distribution of RUL can be obtained analytically.

In order to examine if this is possible, rewrite Eq. 4 as:

$$x(0) = \frac{1}{a^r}(l - \sum_{j=0}^{j=r-1} a^j.b) \tag{6}$$

While $x(0)$ is completely on the left hand side of this equation, r appears not only as an exponent in the denominator but is also indicative of the number of terms in the summation on the right hand side of the above equation. Therefore, it is clear that the relationship between r and $x(0)$ is not linear. Therefore, even if the state variable ($x(0)$) follows a Gaussian distribution, the RUL (r, a realization of R) does not follow a Gaussian distribution. Thus, it is clear that even for a simple problem consisting of linear state models, a straightforward threshold function, and only one uncertain variable that is Gaussian, the calculation of the probability distribution of R is not trivial. Even the distribution type of RUL is unknown for this conceptual problem.

Indeed practical problems considered in the prognostics and health management domain may consist of:

1. Several non-Gaussian random variables which affect the RUL prediction,
2. A non-linear multi-dimensional state space model,
3. Uncertain future loading conditions,
4. A complicated threshold function which may be defined in multi-dimensional space.

It is the goal of a prognostic algorithm to rigorously account for all the uncertain quantities and compute the uncertainty in the remaining useful life prediction. It is important to note that RUL is simply a dependent quantity and needs to be predicted without making any assumptions regarding the distribution type (say, Gaussian) or statistics (say, mean or standard deviation) of RUL. This can be addressed posing RUL prediction as an uncertainty propagation problem (Sankararaman & Goebel, 2013b, 2013a). For this purpose, the remaining useful life prediction needs to be written as a function of all of the uncertain quantities. For instance, in the above conceptual example, Eq. 4 can be rewritten as:

$$r = G(x(0), a, b) \tag{7}$$

Then, the uncertainty in $x(0)$, a and b are propagated through G (note that G is equivalent to solving Eq. 4 for r) to compute the uncertainty in the remaining useful life prediction. In the case of practical problems, such computation is very challenging particular when prognostic calculations need to be performed during the operation of the system.

3.3. Conceptual Algorithm

Given information regarding the state estimate, future conditions, and degradation model, this section further uses a conceptual algorithm for the purpose of illustration. This algorithm calculates the mean and standard deviation of RUL using first order Taylor's series expansion (Sankararaman, Daigle, & Goebel, 2014), and is known as the first-order second moment (FOSM). Note that this simply has been deliberately chosen to illustrate certain pitfalls of existing performance evaluation methods.

For the conceptual example of Section 3.1,

$$\mu_r = G(\mu_{x(0)}, \mu_a, \mu_b) \tag{8}$$

where μ_r, $\mu_{x(0)}$, μ_a, μ_b denote the mean of r, $x(0)$, a, and b respectively. The variance of r, i.e., σ_r^2 can be calculated as:

$$\sigma_r^2 = (\frac{\partial G}{\partial x(0)})^2\sigma_{x(0)}^2 + (\frac{\partial G}{\partial a})^2\sigma_a^2 + (\frac{\partial G}{\partial b})^2\sigma_b^2 \tag{9}$$

where σ_r, $\sigma_{x(0)}$, σ_a, σ_b denote the standard deviation of r, $x(0)$, a, and b respectively.

Typically, $\mu_{x(0)}$ and $\sigma_{x(0)}$ are provided by the state estimation algorithm, and the RUL needs to be predicted by forecasting (extrapolating using the degradation model) the state estimate forward in time; such forecasting is equivalent to the calculation in Eq. 7. For example, consider the following statistics: $x(0)$ follows a Gaussian distribution (with mean and standard deviation equal to 1000 and 200 respectively), a follows a uniform distribution (with lower and upper bounds of 0.990 and 0.995), and b follows a uniform distribution (with lower and upper bounds of -0.005 and 0 respectively). For failure threshold limit $l = 50$, the RUL prediction can be approximated to be a Gaussian distribution based on the above calculation of the FOSM method. The resultant probability density function (PDF) is indicated in Fig. 2.

The various aspects of performance evaluation are discussed in detail using this algorithm. While the above algorithm is simply used for the purpose of illustration, the following discussion can be extended to any type of unit-based prognostic algorithm.

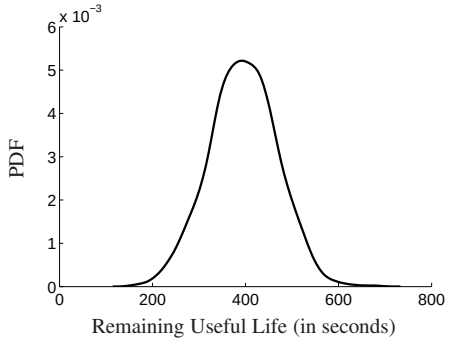


Figure 2. RUL: Conceptual Example

4. CHALLENGES IN PERFORMANCE EVALUATION

Any typical prognostic algorithm uses information regarding the three key elements, i.e., state uncertainty, future uncertainty, and model uncertainty, and computes the remaining useful life prediction. While it would be ideal to compute the entire probability distribution of the RUL, some algorithms compute only certain statistics (like mean and standard deviation) and assume a distribution type (such as Gaussian). Recall that Section 3 stipulated that such assumptions should not be made, and RUL must be fully treated as a dependent quantity.

In order to judge the performance of an algorithm, ground truth data are obtained through experimental studies that *mimic* the various uncertainties that are accounted for, in the prognostic algorithm. Note that it is not individually possible to evaluate how well each of the three key elements have been quantified; only their combined effect on the RUL prediction can be compared against ground truth data.

As far as experiment is concerned, the component/system is at a particular state at any instant of time and there is no uncertainty regarding this state. However, a typical state estimation cannot precisely estimate this state and hence, expresses the uncertainty through a probability distribution. Hence, a typical state estimation algorithm adds extraneous uncertainty, and this would not exist if an idealistic state estimator were present. Similarly, the degradation model uncertainty is also extraneous from the perspective of an algorithm (arises due to the inability to accurately predict the underlying degradation phenomenon), and would not exist if an idealistic, exact degradation model were used. These two types of uncertainty cannot be simulated in a laboratory experiment since they are extraneously added by the algorithm due to the lack of an exact state estimate and an exact degradation model. In fact, effect of state estimation uncertainty and model uncertainty on the difference between the the ground truth and prediction will be equal to zero in the presence of an exact state estimate and an exact degradation model.

However, this is not the case for future loading uncertainty because this uncertainty represents possible future realizations of loading conditions. Hence, it is possible to simulate multiple future loading conditions in the laboratory. However, the challenge lies in the fact that one unit can experience only one set of loading conditions. Multiple loading conditions would have to be simulated on multiple, nominally identical units, and in this case, run-to-failure times of these multiple, nominally identical units will be colored by the inherent variability across them. Hence, it is not possible to experimentally emulate multiple future loading conditions, in the context of condition-based monitoring. And, it is not possible to rigorously evaluate prognostic algorithm performance by considering the simultaneous, joint, effect of state estimation uncertainty, model uncertainty, and future uncertainty on the remaining useful life prediction. Therefore, it is necessary to investigate other practical performance evaluation techniques that can quantitatively judge quality of the remaining useful life predictions of a prognostic algorithm.

5. PRACTICAL PERFORMANCE EVALUATION

This section discusses the most common method of performance evaluation, i.e., comparing the actual run-to-failure time against the algorithm prediction. The shortcomings of this approach are described and new performance evaluation approaches are suggested.

5.1. Ground Truth Comparison

Most existing performance evaluation techniques rely on the availability of the ground truth failure data, and the RUL predicted by the prognostic algorithm can be easily compared against the observed failure time. However, such comparison is not only inequitable, but,sometimes, it may lead to incorrect conclusions.

1. **Inequitable Comparison:** From the time of prediction until the time of failure, the algorithm assumes some uncertainty regarding the future loading and usage conditions. However, the observed ground truth is reflective of only one loading/usage condition that actually happened in reality, thereby implying that similar quantities are not compared. In other words, the experiment contains no uncertainty regarding loading/operating conditions, whereas the algorithm accounted for such uncertainty.
2. **Concluding poor performance of a good algorithm:** The aforementioned inequitable comparison can sometimes lead to concluding that a good algorithm is poor. Consider the case where an algorithm is provided future loading conditions that are completely different from the actual loading conditions. The algorithm may process the provided information accurately and compute the RUL. However, this prediction may be completely different from the observed ground truth RUL. This difference needs to be attributed only to the incorrectly as-

sumed loading conditions and it is not reasonable to penalize the prognostic algorithm in this context. In the context of the conceptual example, the actual loading may have been corresponding to $a = 0.90$ which would have led to a much smaller ground truth RUL than that predicted by the algorithm in Fig. 2. Thus, though the algorithm had been reasonably accurate, its performance would have been judged based on incorrect loading assumptions.

3. **Concluding good performance of a poor algorithm:** Suppose that the prediction of the algorithm is extremely accurate and precise, with respect to the observed ground truth. Then, it cannot be inferred that the algorithm is performing well. This is because the algorithm may not be accurately processing all the uncertainty regarding the future and thereby leading estimates with lesser precision than what the algorithm is supposed to do.

Some of these challenges can be overcome using another type of performance evaluation, as explained in the following section.

5.2. Informed Ground Truth Comparison

It is possible to eliminate the effect of not knowing the loading condition in advance, by waiting until failure. The actual loading/usage condition experienced by the component/system can be observed, and the prediction algorithm can be provided this information. Therefore, the algorithm prediction can be "informed" with the actual loading condition, and the informed-prediction can be computed easily. Note that, at the time of prediction, this information would generally not be available to the algorithm. Therefore, this procedure is only to evaluate the algorithm performance, after eliminating the effect of unknown future loading conditions. All the other information provided to the algorithm need to be reflective of the information available to the algorithm at the time of prediction, such as the state values at the instant of prediction.

In the conceptual example, assume that a component has been run until failure, and the actual loading condition was observed to correspond to $a = 0.994$. Then, the informed prediction can be computed, as shown in Fig. 3. Note that the original prediction has also been shown, for the sake of comparison. This comparison needs to confirm that the observed ground truth falls within reasonable bounds of the informed prediction; note that these bounds are much narrower than the bounds corresponding to the original algorithm prediction.

Similar to the traditional ground-truth-based evaluation, the informed prediction of the algorithm can be compared against the observed ground truth. Note that the former is uncertain because of uncertainty in the state estimate and the degradation model. Note that it is still difficult to evaluate the effects

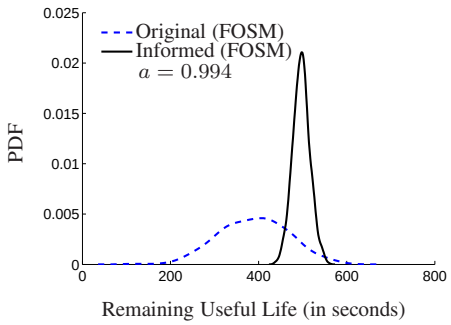


Figure 3. RUL Prediction: Original vs. Informed

of state estimation uncertainty and model uncertainty; in fact, these two quantities could have compounding or canceling effects and such effects cannot be detected and evaluated easily, unless intermediate measurements of the state are available during the experimental set up.

5.3. Assessment of Computational Accuracy

While the above described measures of evaluation focus on characterizing the effects of state estimates, future loading conditions, and degradation model, it is also necessary to check whether the algorithm is accurately processing the different sources of uncertainty. This is not related to accurately predicting the RUL, but is directly associated to the mathematical treatment of the various sources of uncertainty. Some algorithms may average the effect of the different sources of uncertainty on the RUL, and arbitrarily calculate the variance of RUL using approximations and assumptions (Sankararaman & Goebel, 2013b). It is important not to underestimate or overestimate the underlying uncertainty and accurately calculate the probability distribution of RUL. The ideal approach to perform such calculation is the use of Monte Carlo simulation with a large number of samples; though this requires high computational power, this method can be used to check the performance of other algorithms that are suitable for online prediction. In other words, the probability distributions obtained using the specific algorithm and Monte Carlo simulation can be compared and any discrepancy can be quantified, in order to evaluate the performance of the algorithm, from the perspective of integrating the different sources of uncertainty.

For instance, in the conceptual example, if $x(0)$ follows a Gaussian distribution (with mean and standard deviation equal to 1000 and 200 respectively), a follows a uniform distribution (with lower and upper bounds of 0.990 and 0.995), and b follows a uniform distribution (with lower and upper bounds of -0.005 and 0 respectively), then the RUL (defined by Eq. 3, where $l = 50$) can calculated as a probability distribution, using Monte Carlo sampling. Using unit discretization (i.e., the time interval between the k^{th} and $(k+1)^{th}$

instants is equal to one second) for solution, the resultant probability density function (PDF) obtained using exhaustive Monte Carlo sampling (MCS) is shown in Fig. 4. For the sake of comparison, the previously obtained result using FOSM is also shown.

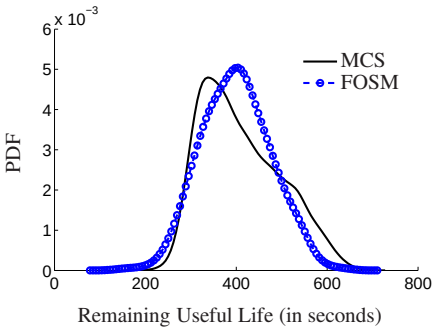


Figure 4. RUL: Conceptual Example

An ideal algorithm should be able to replicate the result from Monte Carlo sampling, as much as possible. A narrower prediction implies that the algorithm is underestimating the total amount of uncertainty whereas a wider prediction implies that the algorithm is overestimating the total amount of uncertainty. The former scenario may lead to unexpected system failure and hence heavy losses, whereas the latter scenario results in extremely conservative decisions and may not use the available resources in an optimal manner.

Note that the FOSM method reasonably agrees with MCS, in this example. This can be attributed to the fact that the the example itself was very simple to begin with. When more uncertain variables are present, and when the degradation model becomes increasingly non-linear, then it is expected that the FOSM result will be significantly different from the MCS result.

5.4. Summary

The search of prognostic performance evaluation measures raises several important questions and concerns. There are four important critical factors that control the performance of prognostic algorithm, and it is not practically possible to individually evaluate the goodness of these factors. While evaluating algorithm performance against observed ground truth seems to be the most widely used method, it is not only unfair but may lead to incorrect conclusions. The informed-prediction method eliminates the uncertainty regarding the future loading conditions, and quantifies the combined effect of state uncertainty and degradation model uncertainty on the RUL prediction. The fourth factor, i.e., whether all the sources of uncertainty are being processed and integrated accurately, can be verified by comparing the algorithm prediction against rigorous Monte Carlo simulation.

An important challenge is the inability to check whether the loading conditions assumed by the algorithm are reflective of what is expected in reality. Is it reasonable to penalize the algorithm for poor performance? Another issue is the ability to identify whether the adverse effect of two (or more) incorrectly estimated quantities jointly cancel out one another, and deceivingly suggest that the prediction is highly accurate and precise. Further research is necessary to address these issues and advance the state-of-the-art in performance evaluation of prognostic algorithms.

6. AN ILLUSTRATIVE EXAMPLE

This section provides an application example to illustrate the various concepts explained earlier in this paper. The example used in this paper predicts end-of-discharge of a Li-ion battery and is borrowed from previous works of the authors (Sankararaman et al., 2014). Since various details about prognostic model development for Li-ion battery are not directly relevant to this discussion they are omitted here, which can be found in (Sankararaman et al., 2014). This example illustrates how one can apply the evaluation method proposed in Section 5 to a real problem. To illustrate pitfalls of raw ground truth comparison and explain the proposed methodology, the rest of this section discusses the various sources of uncertainty in this application example, and explains the previously discussed performance measures.

6.1. Sources of Uncertainty

Consider the prediction of end-of-discharge (EOD) at the initial time instant (t_0). The EOD prediction depends on the following uncertain quantities:

1. **State Uncertainty**: Typically, state estimation is addressed using a filtering technique that can continuously estimate the uncertainty in the state based on the available measurements. In the example discussed in (Sankararaman et al., 2014; Daigle, Saxena, & Goebel, 2012) there are three state variables tracking amount of charge in three capacitive elements of the battery model. These three capacitive elements are referred to as — bulk capacitance (C_b); concentration-polarization capacitance (C_{sp}); and ohmic-drop capacitance (C_s). For complete details of the battery model, and explanation of these terms, refer to (Sankararaman et al., 2014; Daigle et al., 2012).
   It must be noted that in this problem, the charge in C_b is the most influential state variable for predicting the end-of-discharge, and therefore, is considered to be the only uncertain state variable. At the initial time instant, the value of the state variable C_b is denoted by X, and the values of the other state variables are set to zero. Let μ_X and σ_X denote the mean and standard deviation of X.
2. **Loading Uncertainty**: For the purpose of illustration and simplicity, the future loading is assumed to be con-

stant; however, this constant value is chosen at random, and denoted by Y. Let μ_Y and σ_Y denote the mean and standard deviation of Y.

All other quantities are assumed to be completely known constants. The above two sources of uncertainty are sufficient to explain the concepts discussed in this paper.

6.1.1. End-of-Discharge Prediction and Performance Evaluation

It can be seen that the end-of-discharge (EOD) can be written as a function of the uncertain quantities (X and Y), as:

$$EOD = G(X, Y) \tag{10}$$

Note that G is a combination of the degradation model and the end-of-discharge voltage threshold (V_{EOD}) mentioned earlier, and includes all constants that are precisely known. Due to the uncertainty in X and Y, the predicted EOD is also uncertain and represented using a probability distribution. This distribution needs to be compared against experimental end-of-discharge data for performance evaluation. The remainder of this section illustrates various aspects of prognostic algorithm performance evaluation under uncertainty.

6.2. Rejecting a Good Algorithm

If prognostics and prognostics performance are not interpreted and understood correctly, then it may lead to inferring that the algorithm is not performing well.

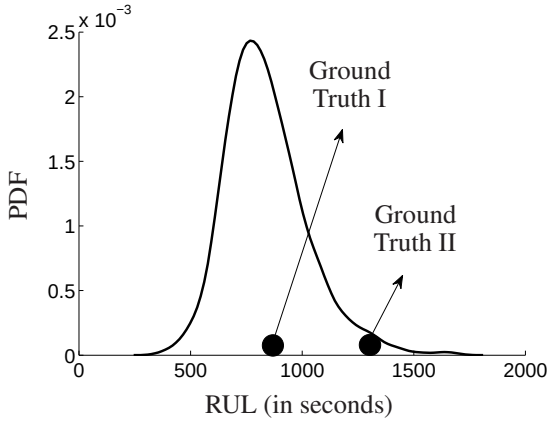


Figure 5. Rejecting a Good Algorithm

For example, consider the RUL prediction (equal to the end of discharge, since the prediction is performed at $t = 0$) in Fig. 5, obtained through Monte Carlo sampling. In this illustration, X and Y are chosen to be Gaussian variables, with $\mu_X = 31115.0$, $\sigma_X = 3111.5$, $\mu_Y = 35$, and $\sigma_Y = 5$. In addition to the RUL prediction, two different ground truth RUL values (Ground Truth I and II respectively) are shown; these two values correspond to different future loading realizations – the more severe results in a shorter life whereas the less severe results in a longer life.

Evidently, the comparison suggests that the algorithm is not performing well since it does not predict Ground Truth II well. However, this may have happened due to several reasons such as:

1. Overestimating the system health during state estimation that leads to the early prediction

2. Overestimating the severity of the loads that leads to early prediction

There is nothing wrong about the algorithm; the information provided to the algorithm is alone questionable. Further, note that the above comparison against the ground truth is unfair since the ground truth represents only one out of several possible realizations considered in the prognostic algorithm.

6.3. Accepting a Bad Algorithm

On the other hand, consider an algorithm that produces the RUL prediction as shown in Fig. 6, and assume that Ground Truth II alone was available through experiments. For example, such an algorithm may compute the RUL in a completely wrong approach in predicting the RUL either by neglecting certain sources of uncertainty or by incorrectly combining the state information along with the degradation model and the threshold model. Therefore, this may lead to concluding that the algorithm is performing well.

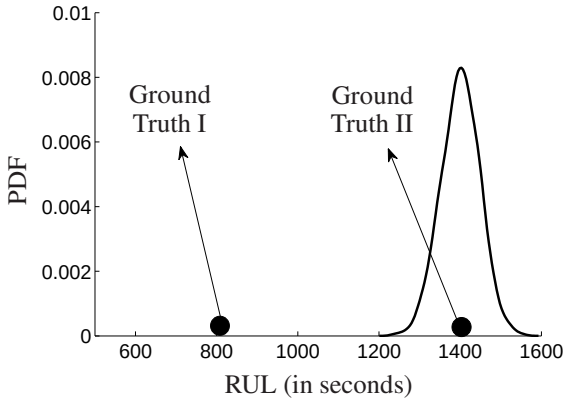


Figure 6. Accepting a Bad Algorithm

However, such a conclusion is incorrect. Since some uncertainty is not accounted for, this algorithm can only capture certain possible realizations of the future but not all possible future realizations; in this case, while Ground Truth II alone be explained by the algorithm, Ground Truth I (which is also a possible future realization) cannot be explained by the algorithm.

6.4. Performance Evaluation

In order to address these issues, this paper discussed two additional measures for performance evaluation. For the purpose of illustration, assume that the FOSM algorithm has been pursued. The first measure of "informed" evaluation measures the actual loading scenario (value of Y, the electrical current, in this numerical example) experienced by the ground truth and "informs" the algorithm with such ground truth. In this case, $Y = 35$ corresponds to Ground Truth I, $Y = 25$ corresponds to Ground Truth II, i.e., a less severe loading leads to longer life. The informed predictions are plotted in Fig. 7, and it can be easily seen that both informed RUL predictions match well with the corresponding ground truth values.

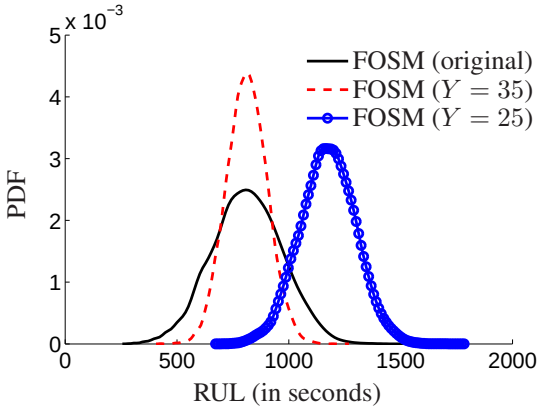


Figure 7. FOSM: Original vs. Informed

The second measure focuses on evaluating the correctness of the algorithm by direct comparison against rigorous Monte Carlo simulation, as shown in Fig. 8. As it can be seen from this figure, the FOSM algorithm is able to capture central tendencies but is not able to capture tail behavior. For this numerical example, the prediction seems to be conservative. However, it could be otherwise for a different set of uncertain quantities and corresponding statistics. That is why it is important to evaluate such correctness by direct comparison against MCS.

6.5. Discussion

Practical problems may have several sources of uncertainty that further complicate performance evaluation through complicated interactions, i.e., Eq. 10 may get complicated with multiple arguments. Many of these sources of uncertainty are "inputs" to the prognostic algorithm, and it is not reasonable to penalize the algorithm if the information regarding these "inputs" are incorrect. That is why it is necessary to develop a rigorous approach to separate (1) evaluation of correctness of information regarding these "inputs" from (2) evaluation of the prognostic algorithm itself. This paper presented a few preliminary steps in this direction and future research may continue to explore the topic of prognostic performance evaluation in further detail.

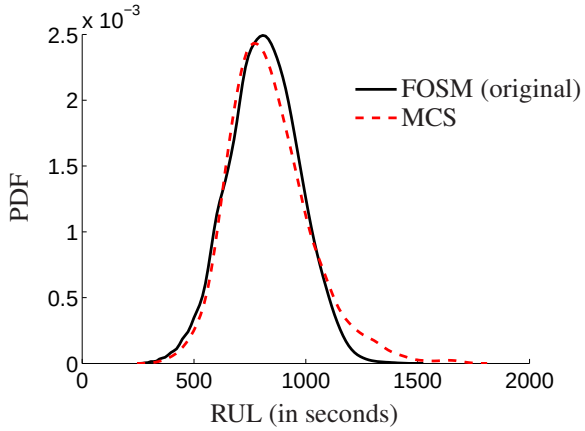


Figure 8. FOSM Algorithm vs. MCS

7. CONCLUSION

This paper discussed the various aspects of performance evaluation of prognostic algorithms in detail, particularly in the presence of uncertainty. To begin with, it was explained that there are several sources of uncertainty that affect prognostics, and that a good prognostic algorithm needs to rigorously account for all of these uncertainties and quantify their combined effect on the remaining useful life prediction. While the presence of uncertainty has been addressed using probability methods, it was explained that the interpretation of probability is not straightforward in prognostics. In testing-based prediction methods, there is inherent variability amongst all the nominally identical specimens that are being tested, and classical statistics-based or frequentist interpretation is applicable. However, in condition-based monitoring, only one unit is studied; therefore, physical variability is absent and all uncertainty needs to be interpreted subjectively. This difference in interpretation plays a key role in understanding the various elements that effectively contribute to the performance of a prognostic algorithm. These elements include: (1) state estimate and associated uncertainty; (2) future loading, operating, and environmental conditions, and associated uncertainty; (3) degradation model and associated uncertainty; and (4) end-of-life threshold and the associated uncertainty. Then, this paper discussed methods for performance evaluation from the perspective of quantifying the combined effect of these elements on the remaining useful life prediction.

First, this paper postulated that it is not possible to evaluate algorithm performance by simultaneously accounting for all these three sources of uncertainty. Second, the most popular technique of comparing ground truth against the algorithm prediction was discussed, and its shortcomings were mentioned. This approach is not only unfair, but also may lead to incorrect conclusions of rejecting a correct algorithm and accepting a wrong algorithm. In order to address some short-

comings of this approach, an "informed evaluation" methodology was proposed; in this method, the true future loading information (available after failure) is provided to the algorithm and then, it is tested whether the ground truth falls within reasonable bounds of the algorithm prediction. Finally, the importance of the mathematical treatment of the different sources of uncertainty was explained, and in this context, it is necessary to compare the performance of any algorithm against Monte Carlo simulation. In other words, given the same information to the algorithm and Monte Carlo simulation, the algorithm prediction needs to be "similar" (in fact, as exact as possible) to that of the Monte Carlo prediction. A narrower prediction implies that the algorithm is underestimating the total amount of uncertainty whereas a wider prediction implies that the algorithm is overestimating the total amount of uncertainty. Future work needs to further explore the concepts of informed evaluation and identify metrics that can express various performance aspects of a prognostic algorithm.

ACKNOWLEDGMENT

The work reported herein was in part funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD) and by the AGSM (Advanced Ground Systems Maintenance) project under the Ground Systems Development and Operations Program in the Human Exploration and Operations Mission Directorate.

REFERENCES

Calvetti, D., & Somersalo, E. (2007). *Introduction to Bayesian scientific computing: ten lectures on subjective computing* (Vol. 2). Springer.

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society.*

DeCastro, J. A. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. In *Annual conference of the prognostics and health management society.* San Diego, CA..

De Finetti, B., & de Finetti, B. (1977). Theory of probability, volume i. *Bull. Amer. Math. Soc*, *83*, 94–97.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008, oct.). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and health management, 2008. phm 2008. international conference on* (p. 1 -6). doi: 10.1109/PHM.2008.4711433

Roychoudhury, I., Saxena, A., Celaya, J. R., & Goebel, K. (2013). Distilling the verification process for prognostics algorithms. In *2013 annual conference of the prognostics and health management society.*

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009, feb.). Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291 -296. doi: 10.1109/TIM.2008.2005965

Sankararaman, S., Daigle, M., & Goebel, K. (2014, June). Uncertainty quantification in remaining useful life prediction using first-order reliability methods. *Reliability, IEEE Transactions on*, *63*(2), 603-619. doi: 10.1109/TR.2014.2313801

Sankararaman, S., & Goebel, K. (2013a). Remaining useful life estimation in prognosis: An uncertainty propagation problem. In *2013 aiaa infotech@ aerospace conference.*

Sankararaman, S., & Goebel, K. (2013b). Why is the remaining useful life prediction uncertain? In *Annual conference of the prognostics and health management society.*

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–17).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1), 20.

Saxena, A., Sankararaman, S., & Goebel, K. (2014). Performance evaluation for fleet-based and unit-based prognostic methods. In *Second european conference of the prognostics and health management society.*

Szabó, L. (2007). Objective probability-like things with and without objective indeterminism. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, *38*(3), 626–634.

Wang, H.-F. (2011, January). Decision of prognostics and health management under uncertainty. *International Journal of Computer Applications*, *13*(4), 1–5. (Published by Foundation of Computer Science)

BIOGRAPHIES

Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks,

system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics, the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence of NASA Ames Research Center, Moffett Field CA. His research focus lies in developing and evaluating prognostic algorithms for engineering systems using soft computing techniques. He has co-authored more than seventy technical papers including several book chapters on topics related to PHM. He is also a member of the SAE's HM-1 committee on Integrated Vehicle Health Management Systems and IEEE working group for standards on prognostics. Dr. Saxena is the editor-in-chief of International Journal of PHM and has led technical program committees in several PHM conferences. He is also a SGT technical fellow for prognostics. He has a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. He has been a GM manufacturing scholar and is also a member of several professional societies for PHM including PHM Society, SAE, IEEE, AIAA, and ASME.

Kai Goebel is the Area Lead for Discovery and Systems Health at NASA Ames where he also directs the Prognostics Center of Excellence. After receiving the Ph.D. from the University of California at Berkeley in 1996, Dr. Goebel worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist before joining NASA. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion and his interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 18 patents and has published more than 300 papers in the area of systems health management.

A Novel Human-Machine Interface Framework for Improved System Performance and Conflict Resolution

Jiaming Li[1] and George Vachtsevanos[2]

[1, 2] *Department of Electrical and Computer Engineering, Georgia Institute of Technology*

jli339@gatech.edu
gjv@ece.gatech.edu

ABSTRACT

This paper introduces a framework for the conceptualization and design of novel operator-aircraft/unmanned system automated interface concepts that will assist to enhance operator reliance on automated advisories. There is a need to explore new human-machine interface strategies stemming from the proliferation over the past years of accidents due to system complexity, failure modes and human errors. Concepts of **autonomy** establish the foundational elements of the work. We pursue a rigorous **systems engineering process** to analyze and design the tools and techniques for automated vehicle health monitoring, human-automation interface and conflict resolution enabled by innovative methods from Dempster-Shafer theory and reasoning algorithms. The emphasis in this contribution is on conflict resolution arising between the human operator (pilot) and on-system automated apparatus. The enabling technologies for conflict resolution borrow from Dempster-Shafer evidential theory, probabilistic and Game Theory for improved system autonomy and reasoning paradigms. The efficacy of the approach is demonstrated via an application to major drive subsystems of a helicopter and an autonomous hovercraft laboratory prototype.

1. INTRODUCTION

There is an urgent need to improve the autonomy, safety, survivability and availability of such critical assets as aircraft and robotic (unmanned) systems that are subjected to internal and/or external threats in the execution of a mission. It has been well documented over the past years that human error is a major cause of class A aircraft mishaps. Moreover, on-board equipment malfunctions, incipient failures and environmental stresses contribute to aircraft accidents. (Hoc, 2000) Most complex systems of interest are now designed and operated with on-board capability to monitor and assess the health of their critical components/subsystems. Such automated processes issue appropriate advisories to the operator/pilot/ground station to take corrective action and avoid detrimental or even catastrophic events. These automated systems and the human operator are invariably exposed to different evidences that result in conflict or disagreement as to the "best" action required to remedy an emergency situation.

A significant challenge for unmanned systems and manned aircraft relates to their ability to resolve conflicts between the human operator and automated advisories, learn from situational awareness cases, and support the operator/pilot in the execution of a mission. It was suggested by an Autonomous Vehicle Operator (AVO) that, at times, "he's been more overcome by the torrent of information pouring in during a drone flight than he was in the cockpit". During the past decades, research has focused on human machine interface issues with an emphasis mainly on the human collecting information and controlling the system. Apparently, the operator is faced with the problem of "information overflow". More recently, with systems becoming more complex and the information processing ability of machines/systems improving, the machine is called upon to perform the same dynamical and automatic functions as those the human was executing in the past. These processes could be affected by uncertainty in the system or the environment. Hence, there is a need to allocate appropriate functions between the human and the machine to reduce the effects of uncertainty.

2. TECHNICAL APPROACH

The Human-Automation Interface-Conflict Resolution and Decision Support-The constituent modules of the human-machine interface architecture pursued in this paper include an on-board automated system that provides to the human operator the most accurate and reliable information regarding the platform's current and future health state through key performance metrics specific to the vehicle and onboard sensors. These are presented to the operator in a

Jiaming Li et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

prioritized manner based on mission essential elements. A modified Dempster-Shafer formula is employed to combine conflicting and incomplete information.

The proposed human-machine interface architecture is illustrated in Figure 1. In the top middle of the figure is the aircraft, the targeted test bed. The pilot or operator is shown on the left. The block under the pilot represents the estimation of current system status. The latter is aided by the knowledge base, which, in return, provides an input to the pilot for emergency actions. Similarly, the Data Acquisition (DAQ) module and aircraft health status estimation block are depicted on the right. There are two major information flows, i.e. information collected by the pilot and the automated system, respectively. The pilot observes current environmental conditions, reads the on-board displays, and communicates with the knowledge base. The Automated System (AS), on the other hand, gathers information from the available on-board sensor suite, represented by the DAQ module. The pilot and the AS apply then reasoning strategies based on the information collected and data/information available in the knowledge base. If there is a conflict between the pilot's decision and the AS's advisory, the conflict resolution module attempts to resolve such conflicts using tools from Dempster-Shafer Theory, probabilistic/fuzzy reasoning paradigms. The final recommendation is generated by the Decision Support System and sent back to the pilot as the final "decision maker" for the "best" action to mitigate the current emergency condition.

Particle Filtering for Fault Diagnosis and Failure Prognosis- The proposed fault diagnosis and failure prognosis framework builds upon mathematically rigorous concepts from estimation theory – an emerging and powerful methodology in Bayesian theory called Particle Filtering that is particularly useful in dealing with difficult non-linear and/or non-Gaussian problems. Particle filtering facilitates the estimation of the state (fault) model over consecutive time instants as measurements become available. The particle filtering routines for diagnosis and prognosis are implemented and executed in near real-time and constitute an integrated framework where the results of diagnosis serve as the initial conditions for prognosis in a transparent and efficient manner.

Fault Diagnosis- The particle-filter-based diagnosis framework aims to accomplish the tasks of fault detection and identification using a reduced particle population to represent the state probability density function (pdf). (Orchard, Wu and Vachtsevanos, 2005)This framework provides an estimate of the probability masses associated with each fault mode, as well as a pdf estimate for meaningful physical variables in the system. Figure 2 shows the anomaly detection results based on an RMS feature. The first plot depicts the progression of the feature as a function of time while the second is the probability of failure; the last one shows the baseline and fault pdfs at 5% false alarm rate. The Type II error is 1.1117% at that specific instant of time.

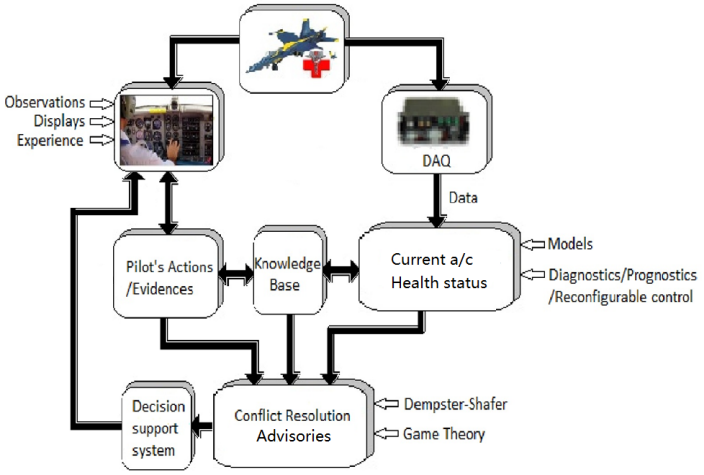


Figure 1. Architecture of human-machine interface

Another performance metric is the Fisher Discriminant Ratio shown at the bottom of the figure.

The "smart" Knowledge Base-A reasoning paradigm called Dynamic Case Based Reasoning (DCBR) that stores cases, matches new cases with stored ones and exhibits attributes of learning and adaptation will be used as the "smart" knowledge base to provide the human operator the ability to interpret automated system outputs correctly and to effectively control the decision making process.

The Pilot/Operator-The pilot/operator, on the other hand, gathers information in a very different way. (Parasuraman & Mouloua, 1996) He/she can exploit a variety of data/information sources, such as displays, alarms - red lights, personal sensing capabilities- the pilot could sense vibrations, temperature rising, noise, etc., visual

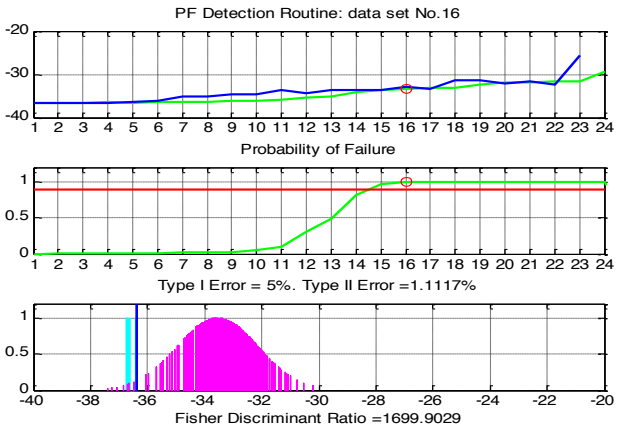


Figure 2. Particle Filtering Routine

observations – look outside the window- rain/ snow, thunder, etc., experience, communication with ground or other aircraft. The pilot gathers information such as oil temperature, fuel pressure, etc. He/she uses this information to assess the current state of the system's health status and to take "initial" actions in the event of an emergency. The operator at this stage may initiate a corrective action or communicate his/her intended actions to the knowledge base. It is understood that timing requirements and sequencing of events in near real-time on-platform are crucial in the final decision making process. The computational requirements burdening the AS are minimized thus allowing for the expedient assessment of the vehicle's state and the application of conflict resolution results.

The Automated System- The Health Management Module-The goal is an advanced integrated reasoning toolset that incorporates justified levels of automated fault accommodation based on prognostic information for enhanced vehicle safety and decision support.

Health and Usage Monitoring Systems (HUMS) acquire on-line in real-time appropriate data and to develop models, algorithms and software that can efficiently and effectively detect faults and predict the Remaining Useful Life (RUL) of failing components with confidence while minimizing false alarm rates. Although the pilot/operator is tasked to use his/her experience, observations and displays to decide on probable causes of an emergency condition and take appropriate initial action, the automated system must perform a series of computationally intensive processes in order to arrive at an advisory for the human operator as to the cause of current adverse conditions and appropriate mitigating strategies. We are introducing a rigorous and verifiable architecture for monitoring and health assessment of critical aircraft systems/components. We outline briefly the major modules of the architecture.

Decision Support System-The decision support system combines these two mass structures derived from the pilot and the automated system using Dempster's rule of combination to arrive at the belief and plausibility for the combined advisory. We are assuming that the final advisory is given to the pilot from the decision support system for action. Moreover, an explanation of how this advisory was derived, i.e. based on what evidence is also provided to the pilot.

3. THE AUTOMATED SYSTEM-PILOT CONFLICT RESOLUTION METHODOLOGY

Conflicts arise between the pilot's intent/commands and automated system commands/advisories. They arise from the different perceptions of the pilot and the automated routines stemming from experience, current data and information available to the pilot and the control architecture which may differ in content, quantity and means for the expedient presentation and follow-up action. The principal task of the Conflict Resolution Module is, therefore, to resolve conflicts between the pilot's actions and those recommended by the automated system.

Conflict resolution is a challenging task that must be addresses methodically in the presence of incomplete evidence, ambiguity and noise. We may apply such methodologies as Dempster-Shafer Theory or Game Theory, among others. In this paper we pursue a conflict resolution method based on Dempster-Shafer theory and specifically Dempster's rule of combination.

Dempster-Shafer Theory-The Dempster-Shafer Evidential Theory is widely used in possibility combination, sensor fusion, artificial intelligence, and conflict resolution areas. (Paksoy & Gokturk, 2011) It allows one to combine evidence from different sources and arrive at a degree of belief that takes into account all the available evidence.

In this formalism a degree of belief, which is also referred to as a mass, is represented as a belief function. Possibility values are assigned to sets of possibilities rather than single events. Dempster-Shafer theory assigns its masses to all non-empty subsets of entities. Application of the Dempster-Shafer Theory requires first and foremost the calculation of the mass functions, as detailed in the sequel.

Assume m_1 and m_2 are two belief function structures on X provided by the pilot and automated system, respectively. m_1 has focal elements $A_1, \cdots, A_k$ and m_2 has $B_1, \cdots, B_p$. We will introduce a modified form of Dempster's rule (Yager, 1987) to combine evidences and avoid counterintuitive results faced by classical methods. Consider two mass functions m_1 and m_2 and define:

$$m = m_1 \perp m_2 \tag{1}$$

where $\perp$ denotes the direct sum and m is calculated as:

$$K = \sum_{\substack{A_i,B_j \\ A_i \cap B_j = \emptyset}} m_1(A_i) m_2(B_j)$$

$$m(\emptyset) = 0$$

$$m(A) = \sum_{\substack{A_i,B_j \\ A_i \cap B_j = A}} m_1(A_i) m_2(B_j), \qquad A \neq \emptyset, X$$

$$m(X) = \sum_{\substack{A_i,B_j \\ A_i \cap B_j = X}} m_1(A_i) m_2(B_j) + K \tag{2}$$

In Dempster's rule, the quantity k is a measure of the degree to which the combined structures disagree with each other. Shafer defines K=log(1-k) as the weight of conflict. So, in Dempster's rule, 1-k represents the normalizing factor needed to assure that the resulting possibility mass satisfies the necessary conditions, i.e. $\sum m(A) = 1$.

Mass function Evaluation- The mass function is the foundation for applying Dempster-Shafer theory to the conflict resolution problem. The estimation of the mass

functions is a challenging problem addressed by several investigators without a satisfactory solution from an analytical and computational perspective. The following sections detail its principal components.

Probability based reasoning-Several assumptions are stipulated for this method: (Basir & Yuan, 2007)

1)There are N types of faults, and M features

2)All features are independent from each other

We employ initially the same formulation as in the previous section.

We use the existing data to fit a two-dimensional normal distribution. In this case, as the two features are independent, ρ is equal to 0. So the distribution now becomes:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}\right]}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}} = f(x)\cdot f(y) \quad (3)$$

Thus, it is written as the product of two independent one-dimensional normal distributions.

For each fault mode, the histogram is generated and then a

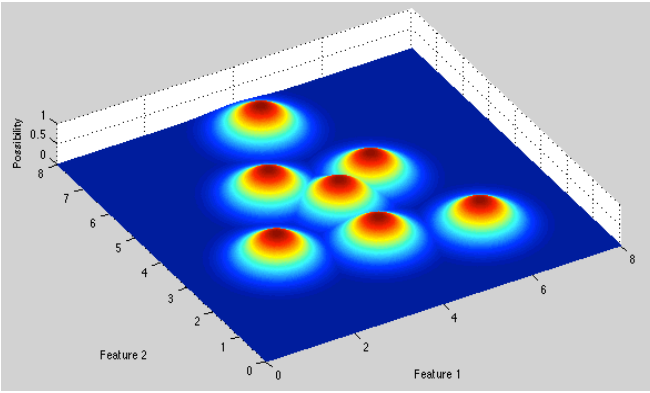


Figure 3. Distributions of fault modes

normal distribution is fitted. Consider next the hypotheses where multiple elements are present. For each hypothesis j we have the label vector L_j. Based on this, the distribution is generated by the following criteria:

$$\begin{bmatrix}\mu_{xj}\\ \mu_{yj}\end{bmatrix} = \frac{\begin{bmatrix}\mu_{x1} & \cdots & \mu_{xM}\\ \mu_{y1} & & \mu_{yM}\end{bmatrix} {L_j}^T}{\|L_j\|_2^2} = \frac{\begin{bmatrix}\mu_{x1} & \cdots & \mu_{xM}\\ \mu_{y1} & & \mu_{yM}\end{bmatrix}\begin{bmatrix}l_1\\ l_2\\ \vdots\\ l_M\end{bmatrix}}{\|L_j\|_2^2} \quad (4)$$

$$\begin{bmatrix}\sigma_{xj}\\ \sigma_{yj}\end{bmatrix} = \frac{\sqrt{\begin{bmatrix}\sigma_{x1}^2 & \cdots & \sigma_{xM}^2\\ \sigma_{y1}^2 & & \sigma_{yM}^2\end{bmatrix}} {L_j}^T}{\|{L_j}^2\|} = \frac{\sqrt{\begin{bmatrix}\sigma_{x1}^2 & \cdots & \sigma_{xM}^2\\ \sigma_{y1}^2 & & \sigma_{yM}^2\end{bmatrix}\begin{bmatrix}l_1\\ l_2\\ \vdots\\ l_M\end{bmatrix}}}{\|{L_j}^2\|} \quad (5)$$

Thus, all the distributions are generated as in Figure 3. For any given states, the actual state vector generated from the sensor suite is represented as: $S = [s_1, s_2, \dots, s_n]$ As in our case, there are only two features, then the S=$[x_0, y_0]$. Define P in a vector form as: $P = [p_1, p_2, \dots, p_{2^M-1}]$

Each element in P is generated by the likelihood S for each distribution:

$$p_i = f_i(x_0, y_0) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\frac{(x_0-\mu_x)^2}{\sigma_x^2}+\frac{(y_0-\mu_y)^2}{\sigma_y^2}\right]},$$

$$i = 1,2,\dots 2^M - 1 \quad (6)$$

Normalizing the P vector, the mass vector is derived by:

$$m_j = \frac{p_j}{\|P\|_1}, j = 1,2,\dots,2^M-1$$

$$M = [m_1, m_2, \dots, m_{2^M-1}] \quad (7)$$

Thus, the mass functions are generated.

We introduce the following Mean Error Bar (MEB) metric:

$$MEB = \int_{t=0}^{t_f}(Pl(t) - Bel(t))dt \quad (8)$$

Or, in discrete form:

$$MEB = \sum_{n=0}^{N}(Pl(n) - Bel(n)) \quad (9)$$

As shown, the belief and plausibility functions give the lower and upper bounds of the possibility function, respectively. The value Pl(t)-Bel(t) stands for the ignorance of the possibility at time t. Usually the possibility is given by the mean of the plausibility and belief functions. If the two values are close, a precise estimate of the possibility function could be given with a small error. Another word, smaller MEB values stands for a more precise estimation. The MEB is, therefore, an appropriate performance metric.

4. THE APPLICATION DOMAINS: HELICOPTER DRIVE SYSTEM

The application domain (in simulation) for the conflict resolution configuration is the Oil-cooler & Intermediate Gearbox (OC-IGB) subsystems of the UH-60 helicopter drive system. The complete drivetrain is shown in Figure 4. The OC-IGB subsystem is highlighted by the red rectangular area. The components include the oil-cooler, the intermediate gearbox, and the tail shaft connecting these components. We define appropriate fault modes and suggest data/observations/displays available to the operator (pilot). On the other hand, we configure the automated system to accomplish sensor data collection and analysis including the

diagnostic, prognostic and control modules introduced previously.

An illustrative example-The proposed human-machine interface framework and the conflict resolution routines may be applied at various levels of the system hierarchy. For example, at the system/subsystem level, the pilot and the automated system may disagree (due to different evidence sources presented to each module) as to which subsystem is experiencing a fault/failure mode. Specifically, may be considering with certain confidence that the Intermediate Gear Box (IGB) of the helicopter's drive system is subjected to a fault. The pilot's conclusion stems from his/her perception/experience, the sensed vibration levels, panel

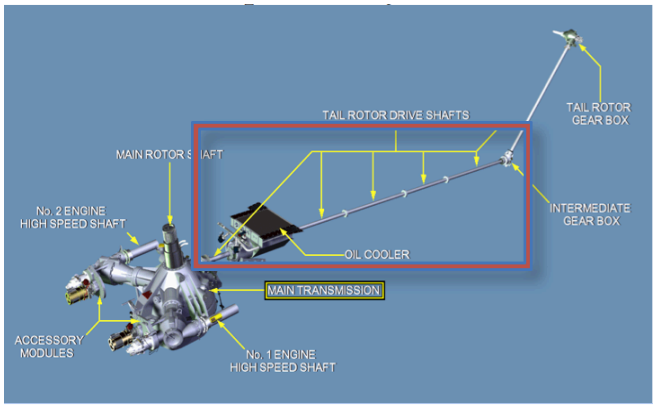


Figure 4. Drivetrain of the UH-60 helicopter

indicators, displays, etc. On the other hand, the automated system is suggesting that the faulty component is the oil cooler. Sensor measurements collected and analyzed by the automated system include oil cooler temperature levels, vibration signals, etc. Shaft coupling in the drive system is one of the main causes for ambiguity/uncertainty corrupting the evidence and resulting in inaccurate allocation of faults. At the component level, the pilot may be surmising, on the basis of the current evidence, that a bearing in the oil cooler assembly is subjected to a fault while the automated system is concluding that the rise in the oil cooler temperature is causing another component to fail.

Features or Condition indicators (CIs) are extracted from the data presented to the automated system. The "best" feature(s) constitute the mass function for the automated system expressed in appropriate probabilistic or fuzzy form. A similar approach is pursued to express the pilot's assertion as a mass function.

We employ the crack level evaluation as a demonstration of the possibility combination for conflict resolution. Consider the crack level as the fault mode for the automated system. We break it down for simplicity into three categories: Light (wear level 0-1inch), Medium (wear level 1-2 inches) and Severe (wear level 2-3 inches). The automated system applies the distance-based algorithm. On the other hand, the pilot senses the vibration in Area 1 (oil cooler) and 2 (IGB). This possibility can be represented as a mass value as well. For instance, the pilot decides: the probabilities of vibration in Areas 1 and 2 are 70% and 90%, respectively. Thus, the possibility of vibration in the oil cooler bearing area is $70\%\times90\% = 63\%$. Based on the Bayesian allocation theorem, this possibility value is allocated uniformly to medium, severe, medium/severe, by:

$$\mathrm{m(Medium)} = \mathrm{m(Severe)} = \mathrm{m(Medium/Severe)} = \frac{\mathrm{p}}{3}$$

So the mass function for the pilot is shown in the Table 1. Then, the decision support system combines these two mass structures using Dempster's rule of combination to arrive at the belief and plausibility functions using the MEB metric, as suggested previously.

Table 1. The mass function for the pilot

Hypothesis	**M(H)**
Light	0.37
Medium	0.21
Severe	0.21
Light/Medium	0
Light/Severe	0
Medium/Severe	0.21
Any	0

It is evident that the combined result decouples the oil cooler bearing and IGB. Meanwhile, it could provide rigorous estimates of the probabilities for each fault mode.

5. RESULTS

The data used in this case study is generated by a MATLAB routine. It consists of sensor values and status evaluations for 37 time indexes. The features discussed above and the status evaluations are extracted from the data set. The pilot's judgment is based on his perception while the Automated System collects the pre-processed data and provides the advisories. Then, the decision support system reads the estimations and gives the combined reasoning result. The simulation procedure is also carried out in MATLAB.

5.1 Oil Cooler Bearing Crack Level Prognosis (Dempster-Shafer Result)

The pilot and the automated system can both do the prognosis based on the information they collected. For instance, in our case, the pilot and the automated system can collect information from time 0 to time 3.2. And based on these information to predict the 3.2 to 15 system situation, as shown in Figure 5.

The upper figure is generated by the pilot and the lower one belongs to the automated system. The lower edge of the figure is the threshold of severe crack. So the pilot believes

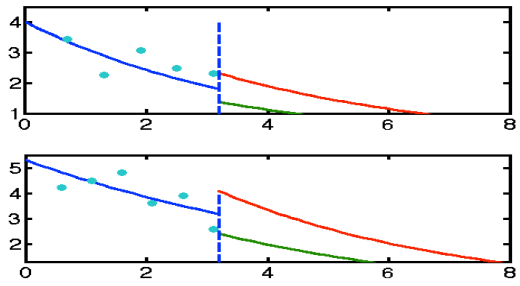


Figure 5. Particle Prediction result by pilot and automated system

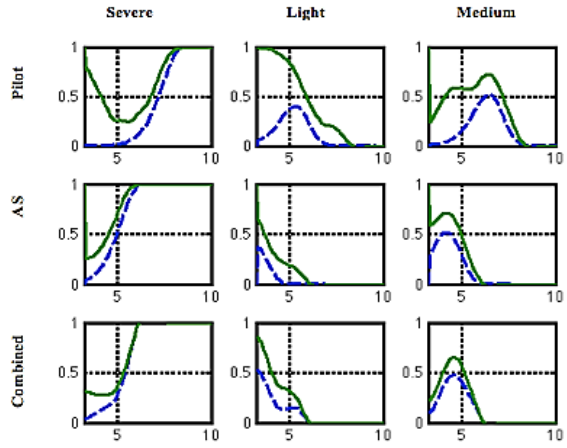


Figure 6. Probability estimated by the pilot, Automated System and the Combined Result

the time for severe crack could be between 4.5 and 6.5. However, the Automated System says the time should be between 5.5 and 8, with a confidence level of 90%. Here comes the conflict between the two reasoning route. So we apply the conflict resolution here to get a combined result, as shown in the Figure 6. In this figure we can see at the time 6 the crack level should be severe with a confidence level higher than 50%. However at time 5 the condition should be light or medium with a confidence level higher than 70%, with is higher than both the pilot and automated system's judgment. This is an example of resolving the conflict.

The MEB is calculated as shown in the Table 2. The table illustrates that the combined result has much smaller MEB than the pilot or AS separately implying that the combined result reduces the risk, or ignorance, significantly.

Table 2. MEB result for each reasoning Routine

MEB	Pilot	AS	Combined
	estimated	estimated	Result
Light	0.2237	0.0805	0.0480
Medium	0.3037	0.0842	0.0751
Severe	0.2152	0.0841	0.0486
Average	0.2475	0.0829	0.0572

5.2 Game Theory Result

First, we map the status evaluation to the action set based on the following table. Here, Action 1 stands for "continue flying" implying that no action is required. Action 2 stands for "prepare to land", which means that maintenance action must be taken after the vehicle reaches its destination. Action 3 stands for "land the aircraft immediately", which means that the aircraft's condition is severe and the pilot must land the vehicle immediately.

Since the automated system monitors the pilot's suggested action(s) automatically, it knows only what action the pilot is taking but not why he takes this particular action and its corresponding probability. Thus, the automated system will evaluate the current status and will estimate the corresponding probability. For example, we are to evaluate the risk for the automated system suggesting Action 1 but the pilot takes Action 3. There are four conditions that recommend Action3 to be taken by the pilot:

Table 3. Conditions which Recommend Action 3

Condition	IGB	Oil cooler bearing	Probability
1	Faulty	Light	$Pr_1 = p_{32} \times p_{21}$
2	Faulty	Medium	$Pr_2 = p_{32} \times p_{22}$
3	Faulty	Severe	$Pr_3 = p_{32} \times p_{23}$
4	Normal	Severe	$Pr_4 = p_{31} \times p_{23}$

Then, referring to the risk table below:

Table 4. Risk Table

Components	Status	Risk for Action1	Risk for Action2	Risk for Action3
Oil cooler bearing Crack	Light	0	0	0
	Medium	16	0	0
	Severe	31	14	0
IGB	Normal	0	0	0
	Faulty	42	17	0

The risk for taking Action 1 is:

$$R_{31} = \sum_{i=1}^{4} Pr_i r_i = 42Pr_1 + 58Pr_2 + 73Pr_3 + 31Pr_4$$

The cost corresponding to each action is estimated as follows:

Table 5. Cost Table

Action	Action 1	Action 2	Action 3
Cost	0	25	50

The cost for taking Action 1 is, of course, zero. The proposed formulation provides thus both cost and risk information. The pilot's suggested action and the AS's advisory are illustrated in Figure 7.

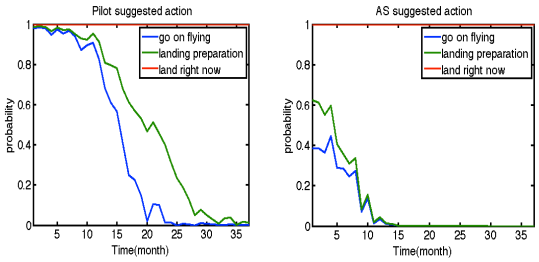


Figure 7 Suggested actions given by the pilot and Automated System

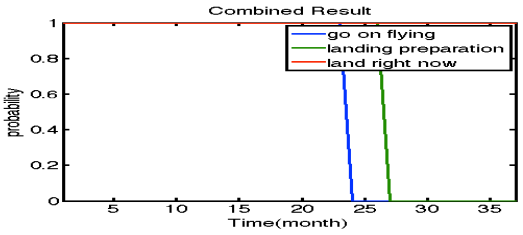


Figure 8 Combined Advisory

Generally, the situation estimated by the automated system is more severe than that of the pilot. Thus, the action suggested by the automated system tends to cost more and is more likely to avoid some severe risks. The combined result, which is the optimum under the given payoff function, is shown in Figure 8.

6. CONCLUSION

In this paper we have described a novel human machine interface framework for conflict resolution. The methodologies applied are modified Dempster-Shafer Theory and Game Theory based conflict resolution methodology. The result shows that the combined result has a better performance than the assessment provided by the pilot or the automated system.

REFERENCES

Basir, O., & Yuan, X. (2007) Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory, Information Fusion 8 (2007) 379–386

Hoc, J.M. 2000, From Human-Machine interaction to human-machine cooperation, Ergonomics, 43:7, 833-843

Lee, T., Richards, J. A., Swain, P. H. (1987) Probabilistic and evidential approaches for multisource data analysis, IEEE Transactions On Geoscience And Remote Sensing, Vol. Ge-25, No. 3, May 1987

Orchard, M., Wu, B. and Vachtsevanos, G., (2005) "A Particle Filter Framework for Failure Prognosis," Proceedings of WTC2005, World Tribology Congress III, Washington D.C., USA, 2005

Paksoy, A., & Göktürk, M. (2011) Information fusion with dempster-shafer evidence theory for software defect prediction, Procedia Computer Science 3 (2011) 600–605

Parasuraman, R. and Mouloua, M. (eds) (1996), Automation and Human Performance: Theories and Applications (Mahwah: Lawrence Erlbaum)

Saha, B. and Vachtsevanos, G. (2006) A Model-Based Reasoning Approach to System Fault Diagnosis, Proceedings of the 10th WSEAS International Conference on SYSTEMS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp64-71)

Yager, R. R. (1987) On the Dempster-Shafer framework and new combination rules, Information Sciences 4, 93-137(1987)

BIOGRAPHIES

Jiaming Li received the B.S degree from the University of Science and Technology of China, in 2011, the M.S. degree from Georgia Institute of Technology, in 2014. Now he is working on his Ph.D. degree in Georgia Institute of Technology.

He is currently an Graduate Research Assistant with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His current research interests include human-machine interface, fault diagnosis and failure prognosis.

George Vachtsevanos received the B.E.E degree from the City College of New York, New York City, in 1962, the M.E.E degree from New York University, New York City, in 1963, and the Ph.D. degree in electrical engineering from the City University of New York, New York City, in 1970.

He is currently a Professor Emeritus of electrical and computer engineering with Georgia Institute of Technology, Atlanta, where he directs the Intelligent Control Systems Laboratory. His work is funded by government agencies and industry. He is the author or coauthor of more than 240 technical papers

Analysis Quality Index - What confidence do you have in your risk analysis?

Leila Salhi[1], Jacek Stecki[2], Chris Stecki[3]

[1,2,3] PHM Technology Pty Ltd, Melbourne, Victoria, 3104, Australia

lsalhi@phmtechnology.com
jstecki@phmtechnology.com
cstecki@phmtechnology.com

ABSTRACT

OEMs and operators of complex mission/safety critical systems are faced with the requirement to mitigate design and performance risks and their economic consequences. A key issue for any engineering organization is the integrity of the analysis that is used to support significant commercial decisions. Analysis outputs used to establish or validate performance criteria should have an appropriately high level of confidence associated with them when entering into significant financial contracts. While risk assessment methods and techniques for analysis are well defined and understood and are captured in various international military and commercial standards, the issue of analysis quality has traditionally been neglected and is not adequately covered in most commercially available engineering analysis tools. The quality of data inputs determines the quality of analysis outputs. A key factor is the source of the parameters used in an analysis. For example input data may be sourced from operational data, or may be based on the engineering judgement of an individual or a third party organization. This paper outlines an approach to analysis quality assessment in a model based engineering environment, focusing on the sources of data and ancillary information to generate an Analysis Quality Index (AQI) for the analysis. The AQI is generated as a dashboard reporting function for the engineering model that is used to provide a confidence rating on the analysis outputs. Analysis Quality Index capability was incorporated into Maintenance Aware Design environment (MADe) software, an integrated tool-set that combines engineering risk analysis capabilities to support systems engineering, design and through-life support.

1. INTRODUCTION

Risk management has become a hot topic over the last decade, its ever increasing application to engineering systems is not always driven by purely technical considerations (Ross, K., & Main, B.W. (2001)). Factors like compulsory compliance with standards (MIL, ISO) and regulation (e.g. FAA), risk of litigation and thus possible audits of the risk assessment process, reliability dependent insurance costs, changes in system management approaches (Product Life Management (PLM), Life Cycle Management (LCM)), changes in sustainment of technical systems (Performance-based Contracts (PBC)), risks to environmental safety etc. cause increased awareness that failures of engineering assets can have penalties.

Operation of an engineering system inevitably leads to system degradation or failure of various degrees, which generate financial, operational (ceased function of the system) and physical risks to assets, human operators or the environment.

To deal with these issues a range of methodologies have been proposed and accepted, especially in the military sector, there are over 150 methodologies dealing with risk management in engineering systems.

The process of risk management is a two-step process:

- Formalized risk identification using various methodologies of risk analysis - Failure Mode and Effects Analysis (FMEA), Failure Mode, Effects and Criticality Analysis (FMECA), Reliability Block Diagram (RBD), Fault Tree Analysis (FTA), etc. see International Standards Organization (ISO) (2004).
- Risk elimination by changes in system design, maintenance, operation etc.

Of course we must remember that risks are assessed and dealt with during the design process, albeit not necessarily using formalized methods.

The objective of risk identification is to determine how the system may fail, and how such failure affects system safety, performance, availability, etc. Analysis provides metrics of

Leila Salhi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

risks (e.g. criticality, reliability) which are the basis for corrective actions (e.g. design changes or changes in maintenance procedures). The formalized risk identification, depending on how and when it is applied, has varying impact on risk reduction. Ideally, it should be concurrent with design of the system so risks identified during design process can be eliminated and/or minimized by modification to design. This approach is optimal in terms of cost, time and degree of risk reduction.

However in practice, formalized risk identification (FMEA, RBD, FTA) is not conducted concurrently with design or is carried out too late to accommodate design changes. In these circumstances, risk analysis has only limited impact on the system design and is often conducted at completion of the design process to generate contractual deliverables or achieve compliance.

The 'concurrent with design' approach is also not possible when dealing with legacy systems. In the case of such a system, we may only use workaround solutions to mitigate risk (better maintenance, sensing) as design changes are often not feasible or possible. Methods like Reliability Centered Maintenance (RCM), Maintenance Effectiveness Review (MER) and Back-fit RCM are used to determine maintenance practices which can reduce operational risk. These methods often lead to outcomes such as Condition Based Maintenance (CBM) and Prognostics and Health Management (PHM).

With a growing importance of risk management methodologies, the quality of the methods is becoming important. Low quality of risk assessment may increase rather than decrease the cost of designing and operating of technical systems.

According to a Google search, the topic of quality of risk assessment is very prevalent - 60,000k results for "risk analysis engineering" and 81,200k results for "quality of risk analysis" engineering – it is currently seen as an important attribute of risk management. Table 1 presents the most widely used methods of risk assessment:

2. THE PROBLEM – CURRENT APPROACHES THAT IMPACT THE QUALITY OF RISK ANALYSIS

The current industry approaches to support risk analysis are primarily database or spreadsheet based software. The use of such software to conduct the required analysis generates a number of significant issues in terms of the cost of conducting analysis, quality of the analysis, system level analysis and scheduling (Bednarz & Marriott (1988), Kara-Zaitri C., Keller A., Barody I. & Fleming, P. (1991), Ormsby A., Hunt J. & Lee M. (1991). The main factors impacting the quality of analysis are the quality and quantity of data used.

- Limited knowledge capture / reuse

Spreadsheets are an obstacle to knowledge transfer which impacts the quantity of data available for risk analysis. The fact that spreadsheets can normally only be updated by the people that created them, is also critical to ensure maximum coverage of the risk analysis. Spreadsheets are not easily configuration managed based on operational data or as changes in the platform are made. Furthermore, the results of a performed analysis cannot be automatically transferred and used to support related analysis methods.

Table 1. List of the most widely used methods of risk assessment according to Google search results (June 2014)

Method of risk assessment	Quantity
FMEA	3,270k
Reliability Diagram	20,600k
Fault Tree	30,000k
Fault Analysis	45,200k
Failure Analysis	113,000k
Performance Based Contract	70,200k
Engineering Risk Audit	34,400k
Condition Based Maintenance	16,700k

- Inconsistency of terminology

The quality in the analysis is significantly impacted by the lack of industry wide taxonomies to define functions and failure concepts, which brings issues of ambiguity and inconsistency of terminology. Risk analyses are also artefact driven (based on attributes of the platform) and performed on a specific state of the system. A snapshot of the system is thus captured by the analysts in spreadsheets and the designer is rarely involved in all iterations of the analyses - this can lead to poor data quality that is used in the risk analysis.

- Retrospective analysis

Usually analysis is done retrospectively (rather than concurrently) at the end of the design process using spreadsheets/database FMEA/FMECA mainly to document the outcomes for compliance or contractual requirements. Evans J. (1992) in his editorial wrote "..The idea that all the experts and number-crunchers should come in after a design was virtually complete, and second-guess the designers was stupid to begin with..".

- Disparate models

Industry practice usually relies on the usage of disparate models of a platform and its Bill of Materials (BOMs) that reside within the functional stovepipes of an organization. This is an obstacle for comparing and controlling the data. Inconsistencies in models such as holes in the BOMs or in the structure of the system may cause coverage losses that are not obvious using spreadsheets.

- Bottom-up – inductive approach.

Current methods to conduct risk analysis are inductive (based on brainstorming) and use a bottom-up approach. It is therefore difficult to visualize and aggregate all the data in order to analyze a system in whole. Each piece of the system data is stored by each stakeholder in spreadsheets. This implies a suggestive process to support the risk analysis process as the assumptions underlying analysis, data sources and knowledge of thought processes of the team members are generally not recorded. As a result, the quality and coverage are affected: a bottom-up approach may result in comments being missed (coverage) and missing the source of the data (brainstorming).

- Subjective analysis audit

Various FMEA guides/books stress the importance of FMEA quality see Carlson. C. S. (2012) and McKinney B. (1991). However, the FMEA quality audit is rather subjective as it relies on subject matter expertise and often is limited to checking that the standard procedure was correctly followed. This does not provide accurate and objective assessment of the quality of analysis. A major problem is repeatability of FMECA when carried by a different team of analysts (Bell D., Cox L., Jackson S. & Schaefer, P. (1992)).

- Platform reliability based on design parameters

In current engineering practices, designers do not necessarily understand how the operators will use the system and this is a critical issue for the reliability of the platform as (Reliability, Availability and Maintainability) RAM / (Integrated Logistics Support) ILS should be based on operationally determined RAM parameters rather than the design parameters. Design parameters are normally sourced from third party references that do not account for concept of operations, environment, etc. Thus it is important to document the source of the information, and list associated assumptions or else quality issues will occur.

- Isolated system analysis

Historically individual technical risk assessments associated with the deferral of maintenance or acceptance of technical defects are conducted in isolation using spreadsheets and therefore do not take into account the potential dependencies across the platform. This could lead to either safety issues or equipment breakdown and thus additional efforts to mitigate risk. Integrating isolated analysis on the higher system level by merging different spreadsheets is almost impossible due to potential taxonomy and hierarchy issues. This impacts the quality of the aggregated analysis performed at the system level.

3. MAINTENANCE AWARE DESIGN ENVIRONMENT (MADE)

MADe (Rudov-Clark S., Stecki J. & Stecki C. (2011)) is a model-based engineering software tool for conducting risk assessment (FMECA, RAM, RCM, FTA) – where each element in the model is associated with a number of key attributes such as its functional description, the specific physics of failure information (cause, mechanism, fault, symptoms) – as shown on Figure 1- and their relevant criticality based on the system performance requirements.

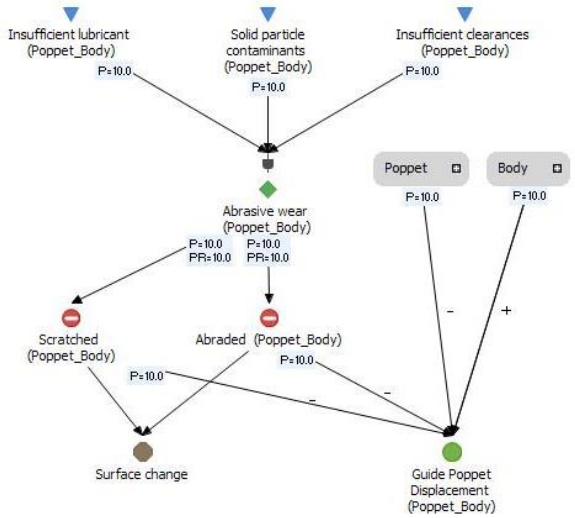


Figure 1. MADe Failure diagram - mapping of failure concepts

MADe utilizes simulation to propagate and trace the dependencies and impacts of any fault injected into the system as shown on Figure 2. This data is used to generate a functional risk assessment based on the associated physics of failure. Simulation is an important feature of the tool, as with highly complex systems it is difficult to identify how the impacts of a failure will propagate – without this knowledge it is impossible to accurately determine the criticality of a specific failure mode.

MADe automates the dependency mapping of a system using the functional path propagations that are generated in the model. The system model is easily updated, modified and MADe enables to conduct 'what-if' analysis for an actual or proposed design and its constituent systems, components and parts.

As it is simulation based, the software is fundamentally and significantly different from spreadsheet/databased tools because the model and therefore the analysis is extensible, objective and repeatable. As a Model Based Engineering (MBE) tool, MADe offers a number of advantages over available spreadsheet/database FMEA/RAM toolsets.

- Knowledge capture, reuse and transfer

All knowledge about the system and its components is captured in models which can be saved and reused for any other project. These user developed models are stored in a re-usable directory called a Library.

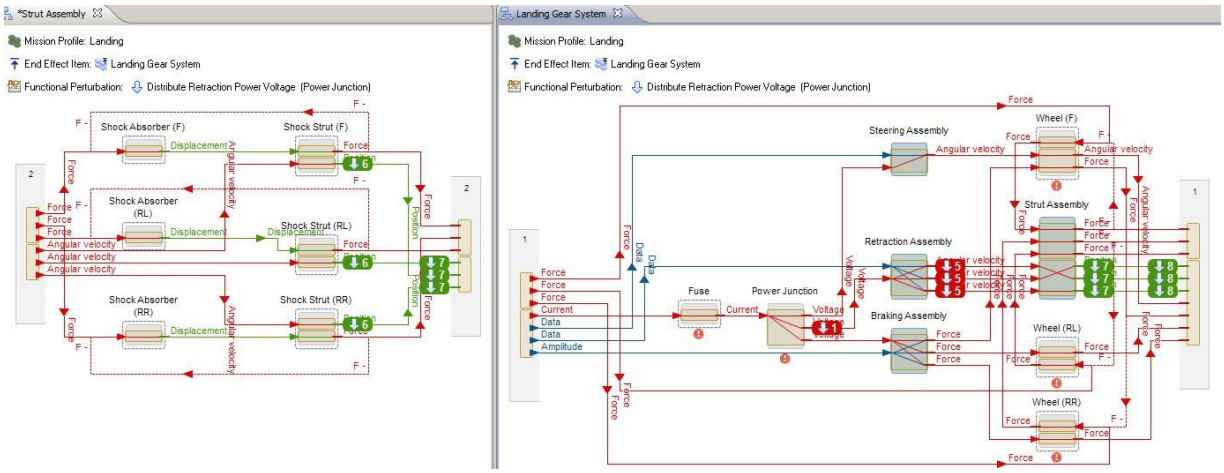


Figure 2. MADe functional diagram of landing gear - showing failure propagation

Models of components/systems can be loaded from the Library and re-used to represent a new system (dependencies will be automatically established). The key benefit is the improved quality of analysis, as knowledge is captured and re-usable for future projects.

- Standardized taxonomy

MADe uses standardized taxonomy of functions/failure concepts to ensure that there is consistency of terminology (and therefore understanding) within the organization and currency of data at each stage of the platform life-cycle (Rudov-Clark, S.D. & Stecki, J. (2009).

Audit and validation are based on the input of references for the sources of data. A standardized taxonomy brings objectivity in the performed analysis.

- Concurrent engineering

Model-based Engineering (MBE) enables concurrent engineering features such as functional simulation which means that the development of a system model can be associated with the functional requirements of a system rather than a specific design. This enables the ability to generate the model - and conduct modelling analyses - at the conceptual stage of the design process to evaluate the impact of changes to the design and mitigate risk at an early stage in the platform life-cycle.

- Integrated capabilities

MADe uses a single model (a *Single Source Of Truth (SSOT))* as basis for other analysis tasks. A model of the system is used for reliability analysis (both functional and hardware), sensor selection (sensors coverage), Reliability Centered Maintenance (RCM) etc. This eliminates the need to export data or results of analysis as the same model is used for all the analysis.

- Configuration management of the analysis

Because MADe generates each analysis based on the common system model, the impacts of any changes made by other functional groups within the organization are automatically reflected in the model (and thus future analyses). This considerably improves the quality of analyses as data come from a *SSOT* model.

- Integrated system analysis

The toolset uses automated dependency mapping which eliminates the manual determination of the impacts of failures across the system. This enables risk analysis to be based on objective and verifiable data. MADe automatically establishes these connections and updates them when the system model is modified. This is a major benefit for increasingly complex and integrated systems. The level of details and dependency mapping enable risk identification at the platform level down to the component level leading to enhanced traceability of data.

- Dependencies mapping

A functional model represents a flow of energy, material or signal in the system. Based on *(SSOT*) model of the system, functional relationships and failures/effects dependencies in a system for both functional and physical failures are defined using standardized taxonomies.

- What if..” and “As is..” analysis

“What if...” analyses are often focused on the rearrangement of connections between models and/or inclusion of different components. This capability is normally too time consuming to be achieved using a database approach, but can be expedited using a MBE approach (e.g. copy-paste and library re-use) leading to otherwise unachievable options.

MADe has the ability to update the parameters in the model based on operational data in order to conduct analysis of the system based on an 'as-is' performance state rather than 'expected' (design) state. This has a significant impact on the supportability posture for equipment.

- Objective analysis audit

An objective approach to conduct risk analysis is beneficial for audit purposes and quality checking. A good example of efficient risk analysis verification is FMECA. Using an AQI, the analyst can easily check the completeness of the analysis based on the quality and quantity of the data inputs. When it comes to project management, an AQI can provide a means to evaluating the confidence level of a system globally or a particular risk analysis in order to validate a project.

- Effective integration with the organization IT architecture (specifically PLM).

Current challenges in PLM consist in using a single point of truth for the RAM / ILS analysis that can be shared by the design / supportability engineering communities. As a simulation based model, MADe offers the ability to configuration manage the associated ILS analysis and outputs for a system and automatically regenerate the artefacts that result from any modification to the design or changes to the maintenance regime.

4. ANALYSIS QUALITY ASSESSMENT

For any analysis or simulation based analysis, poor quality inputs or improperly defined scenarios create meaningless results. How then to assess the quality of risk analysis?

Analysis Quality Index (AQI) is the process of determining that an analysis provides a correct outcome or solution. An AQI may be applied to numerous different analyses or algorithms (e.g. FMEA, Criticality, Reliability) to evaluate and document the accuracy of the results. An AQI process is implemented in MADe to increase data quality and enable objective audit of risk analysis. The main function of an AQI is to enable the modeler to capture the assumptions used during the process of creating the model. A work flow assessing an AQI is shown in Figure 3. In MADe the process starts with setting up annotation policy Figure 4.

The findings from an AQI can be used to document an analysis or query the effectiveness of another analysis. An example of this is performing an AQI on a FMEA to determine the confidence of a particular subsystem, which when integrated to the system level can identify high-risk areas in a project.

When carrying out engineering functions, assumptions may not be listed, or listed after the fact leading to poorly documented work.

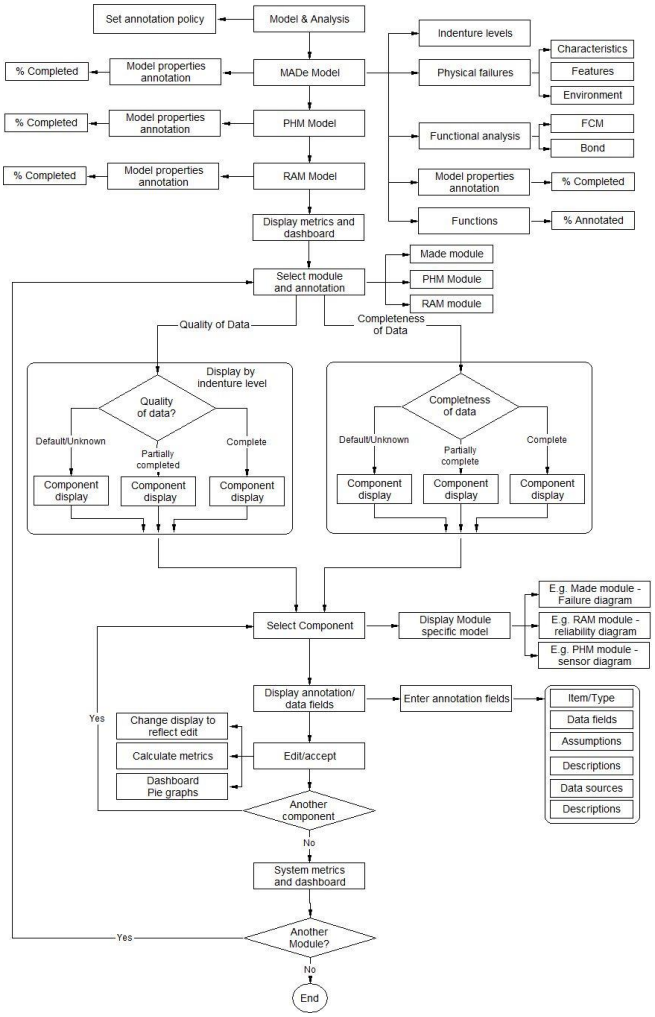


Figure 3. AQI workflow implemented in MADe

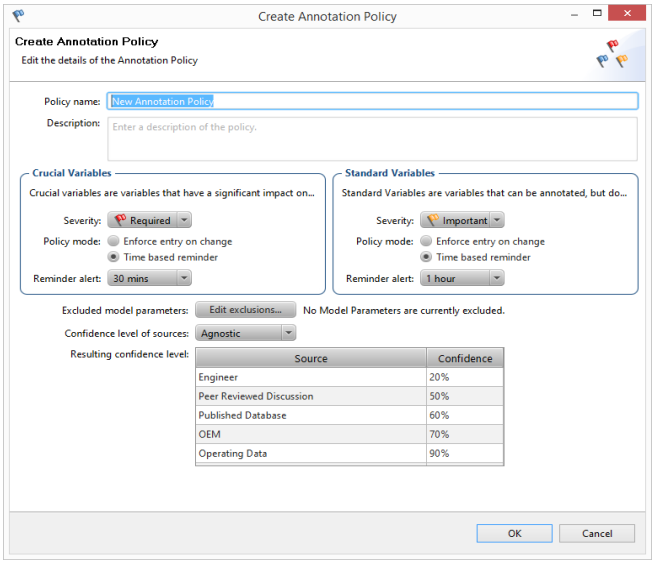


Figure 4. Annotation policy setting

The quality of the assumptions, data and parameters used in a model directly affects the integrity of any analysis output. The solution for this issue in MADe a user enters assumptions for each piece of data, Figure 5.

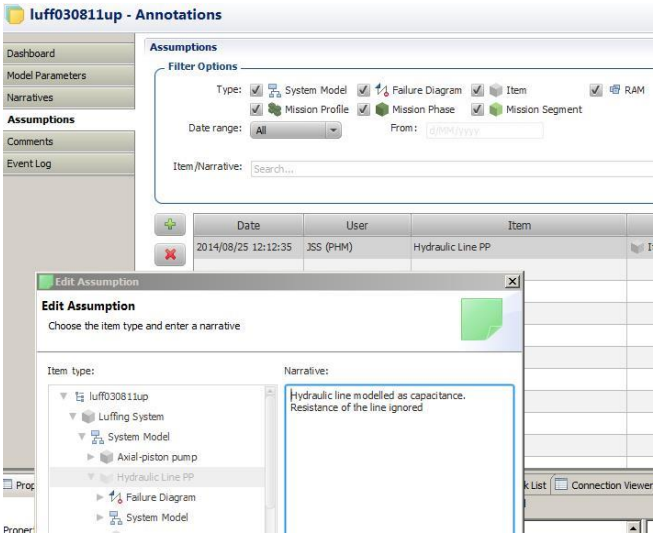


Figure 5. Editing/entering assumptions

However, it is difficult to keep track of the data sources and assumptions that support any parameter used in a model, particularly if multiple stakeholders (including departments, groups, teams and external suppliers) are involved in system development. Therefore a structured approach to documentation and assessment of data quality is essential.

Considering the evolutionary nature of a model, it becomes necessary to capture this information concurrently as the user is modelling. Using this facility will allow more accurate models based on listing of the relevant assumptions, detailed entries including narratives and more consistent processes by capturing considerations. Shown in Figure 6, each parameter edited or changed in the model can be tracked and assessed using an annotation feature that requires each stakeholder to document his data.

To summarize the data quality assessment of a model-based risk analysis such as FMEA/FMECA, requires evaluation of two key metrics:

- Completeness of Data (Data Coverage)
- Data Quality

Once those two metrics are assessed, they can be aggregated to determine the overall confidence level of a particular risk analysis or completeness of a model. An AQI becomes increasingly important as the analysis or models become more complex, thus requiring greater control and management of a larger set of data. The quality assessment concept is especially beneficial for model-based risk analysis.

4.1. Data Coverage

The AQI is a metric that may be used to determine the completeness of data used in the analysis. Missing data regarding the system can result in poor coverage of the risk analysis, especially in a complex analysis where there are numerous inputs required. If any of these inputs are missing then the completeness of the analysis is weakened. Completeness can be considered as the ratio the amount of data entered / the amount of data required. Therefore if all data for a process/analysis is entered then the completeness would be 100%, providing a high confidence with the process/analysis. A higher completeness will improve confidence during an audit and prove better traceability of the analysis. Although it is important to note that while an analysis/process is complete, it may not be high quality.

4.2. Data Quality

Data quality involves documenting the source, confidence level and assumptions underlying each piece of data that is used as an input parameter for the analysis.

This process aims at documenting critical questions regarding a model or a particular analysis:

- Where does a particular parameter or data set come from?

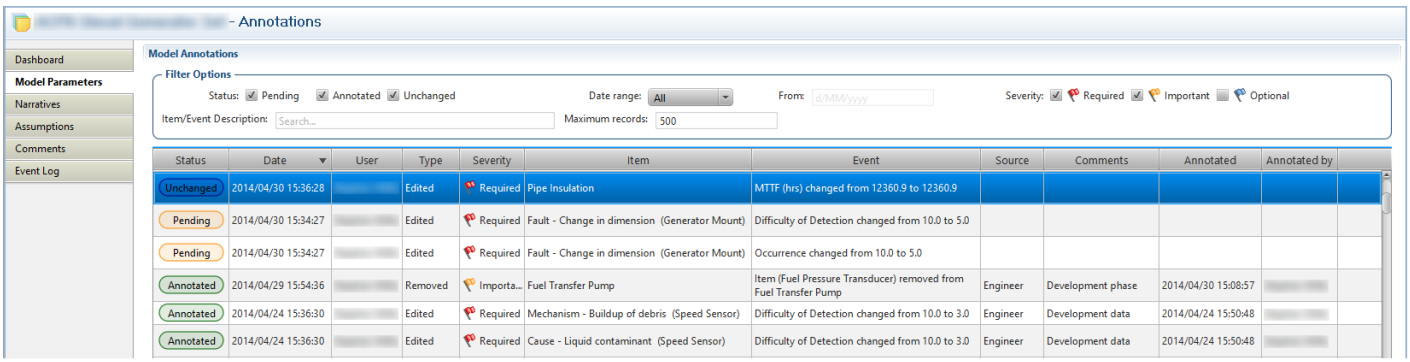


Figure 6. Annotation summary

- Who sourced this data?
- Why was it set to this particular value?
- Which confidence to assign to a particular data?

The quality of data can range from conceptual (brainstorming) to collected data (operation) and is important in defining the quality of the data used in the analysis. Previous articles on the quality of analysis (Evans J. (1992).) explain that in order to avoid poor data quality, "it is essential for everyone with a real-word problem to insist on an adequate, numbered, list of assumptions, where the assumptions are in reasonably plain language". To rank quality, different categories can be assigned which correspond to different sources (e.g. engineer, database, etc.). By defining a data source type, a confidence level can be assigned to each type which may be aggregated to provide an overall level of confidence. As the quality of the data sources increases so does the quality of the analysis. The categories and weightings of sources can be adjusted for specific environments or applications. It is also important to track the source where data is obtained from, note the source of the information, time/date of data entry and allow annotation of a particular entry. This information is automatically updated as data is being annotated in the model to provide the percentage of annotated data, data quality, as well as an overall confidence level in the model as shown in Figure 7 and Figure 8.

Figure 7. Coverage, quality and confidence level

5. CONCLUSION

This paper has outlined a unique approach to assess the quality of risk analysis in a model based engineering environment. In current industry approaches, the extensive usage of spreadsheet/database based tools to conduct risk analysis generates a number of significant issues in terms of cost of conducting analysis, quality and objectivity of the analysis, as well as system level analysis. To solve those issues, it is essential to conduct data quality assessment focusing on the quality and quantity of data used as parameters in the analysis. A good example of assessing the quality of analysis is to apply data quality assessment to model-based risk analysis. The quality assessment process implemented in the MADe software provides objective auditability of all relevant information regarding a particular analysis or a whole system. The confidence level in analysis outputs and thus the quality of analysis are optimized by:

- Documenting and reviewing all parameters used in the model / analysis.
- Mitigating posting cycle issues as expert knowledge to a project file is retained.
- Ensuring that all relevant supporting assumptions are captured.

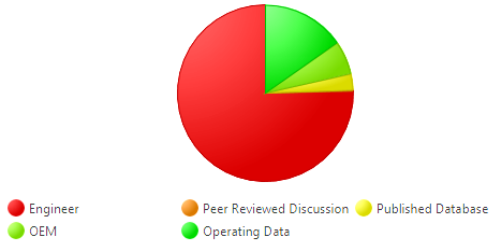


Figure 8. Pie chart showing origin of data

6. FUTURE WORK

While this paper has focused on presenting the application of data quality assessment to a model-based risk analysis (AQI) there are other possible applications of data quality assessment.

- Model Quality Index (MQI)

This is the process of assessing the manner and degree to which data used in a model is an accurate representation of the real world and of establishing the level of confidence of this assessment. This index would be useful in model or simulation environments to determine the validity and correctness of a model compared to the system it is based upon. The findings from an MQI could be useful in learning how to create a more accurate or correct model of a system.

- Process Quality Index (PQI)

This is the process of assessing the confidence and adherence to a particular workflow or process. This could be applied to an engineering process and used to assist learning of a new process or even the audit of an existing process within a company. Findings from a PQI could be applied back into the process to optimize it for its function within a company.

REFERENCES

International Standards Organization (ISO) (2004). ISO 31000:2009: Risk Management - Principles and Guidelines. Genève, Switzerland: International Standards Organization

Bednarz, S. and Marriott, D. (1988). Efficient Analysis for FMEA. *Proceedings of the 1988 IEEE Annual Reliability and Maintainability Symposium* (416-421), January 26-28, Los Angeles, CA, USA.

Bell D., Cox L., Jackson S. & Schaefer, P. (1992). Using Causal Reasoning for Automated Failure Modes and

Effects Analysis (FMEA). *Proceedings of the I992 Reliability and Maintainability Symposium* (343 – 353), January 21-23, Las Vegas, NV, USA.

Carlson. C. S. (2012). *Effective FMEAs.* Hoboken: John Wily & Sons, Inc.

Evans J. (1992). Editorial - CBIA. *IEEE Transactions on Reliability*, vol. 41, 1. March 1992.

Evans J. (1992). Editorial - The Demise of R&M. *IEEE Transactions on Reliability*, vol. 41, 1, . March 1992.

Gilchrist, W. (1993). Modelling failure modes and effects analysis. *International Journal of Quality and Reliability Management*, 10(5), 16-23.

Hunt J.E., Pugh D.R. & Price C.P. (1995). *Failure Mode Effects Analysis: A Practical Application of Functional Modeling.* Applied Artificial intelligence, 9(1), 33-44.

Kara-Zaitri C., Keller A., Barody I. & Fleming, P. (1991), An Improved FMEA Methodology. *Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium* (248-252), January 29-31, Orlando, FL, USA.

McKinney B. (1991). FMECA, The Right Way. *Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium* (253-259), January 29-31, Orlando, FL, USA.

Ormsby A., Hunt J. & Lee M. (1991), Towards an Automated FMEA Assistant. In Rzevski G. & Adey R. (Eds), *Applications of Artificial Intelligence in Engineering VI* (739-752), Southampton, Boston: Computational Mechanics Publications.

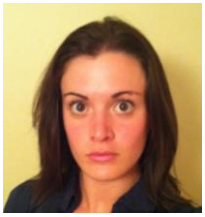
Ross, K. & Main, B.W. (2001). Risk Assessment and Product Liability. *Product Liability Committee, 34-38.*

Rudov-Clark, S.D & Stecki, J. (2009). The language of FMEA: on the effective use and reuse of FMEA data. *Sixth DSTO International Conference on Health & Usage Monitoring*, March 9-12, Melbourne, Australia.

Rudov-Clark S., Stecki J. & Stecki C. (2011). Application of advanced failure analysis results for reliability and availability estimations. *AERO '11 Proceedings of the 2011 IEEE Aerospace Conference* (pp 5), March 5-12, Big Sky, USA.

Wirth, R., Berthold, B., Kramer, A. & Peter, G. (1996). Knowledge-based Support Analysis for the Analysis of Failure Modes and Effects. *Engineering Applications of Artificial Intelligence*, 9(3), 219-229.

BIOGRAPHIES



Leila Salhi is a junior engineer at PHM Technology. She studied physics engineering and micro-technology in a French engineering school and completed an associate's degree of management of high-tech innovation. She worked on a research project involving an Australian mining company during her internship at one of Royal Melbourne Institute of Technology University's research centers.

Dr. Jacek Stecki is the Chief Technology Officer of PHM Technology. He has over 40 years research and industrial experience working in Poland, Australia, USA and Italy, Norway, UK, Switzerland and Denmark. Research projects in fields of fluid power, risk management, machine condition monitoring and modelling and simulation. Director of the Centre for Machine Condition Monitoring, Monash University, Australia (1980-1998). Consultant in Australia, USA, Norway, Switzerland and Brazil with major industrial and mining companies in areas of fluid power, subsea engineering and machine condition monitoring.. He has also written over 120 technical and scientific papers on the subjects of maintenance aware design, fluid power, simulation, risk and reliability assessment (MADe software), model based diagnostics, tribological systems (contamination/wear).

Chris Stecki Chris Stecki is the CEO and Co-Founder of PHM Technology, the developer of the Maintenance Aware Design environment (MADe). MADe is currently used by Defence organisations and their suppliers in Australia, Europe and the US. Chris is a frequent presenter at industry and technical conferences around the world on engineering system design and supportability (particularly System Health Management). He has co-authored a number of technical papers relating to aspects of the engineering design process, and the advanced engineering and IT techniques used in the on-going development of the MADe software.

Challenges in Concrete Structures Health Monitoring

Sankaran Mahadevan[1], Douglas Adams[2], David Kosson[3]

[1,2,3] *Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN, 37235, USA*

sankaran.mahadevan@vanderbilt.edu
douglas.adams@vanderbilt.edu
david.kosson@vanderbilt.edu

ABSTRACT

Structural health monitoring needs to produce actionable information regarding structural integrity that supports operational and maintenance decision making that is individualized for a given structure and its performance objectives. An effective Prognostics and Health Management (PHM) framework for aging structures (subjected to physical, chemical, environmental, and mechanical degradation) needs to integrate four elements – damage modeling, monitoring, data analytics, and uncertainty quantification. This paper briefly discusses available techniques and ongoing challenges in each of these four elements of PHM, in the context of concrete structures. A Bayesian network approach is discussed for integrating heterogeneous information from multi-physics computational models of degradation processes, full-field measurement techniques, big data analytics, and various data and model uncertainty sources. Such a comprehensive framework can quantitatively support decisions regarding appropriate risk management actions.

1. INTRODUCTION

The purpose of structural health monitoring is to provide information to the decision-maker in a manner that is suitable for risk management with respect to structural integrity and performance. Risk management decisions include sustainment decisions regarding inspection, maintenance and repair, as well as operational decisions regarding the mission demand limits for the system and its operating conditions. In all engineering systems, such decisions are made in the presence of uncertainty that arises from multiple sources. The various types of uncertainty include natural variability (in loads, material properties, structural geometry, and boundary conditions), data uncertainty (e.g., sparse data, imprecise data, missing data, qualitative data, and measurement and processing errors), and model uncertainty (due to approximations and simplifying assumptions made in diagnosis and prognosis models and their computer implementation). An important challenge is to aggregate the uncertainty arising from multiple sources in a manner that provides quantitative information to the decision-maker about the future risks for structural integrity and performance, as well as the risk reduction offered by various risk management activities, thus facilitating quantitative risk-informed cost vs. benefit decisions.

The information available in structural health monitoring is quite heterogeneous, since the information comes from a variety of sources in a variety of formats. The heterogeneous sources include mathematical models, experimental data, operational data, literature data, product reliability databases, and expert opinion. In addition to the specific system being monitored, information may also be available for similar or nominally identical systems in a fleet, as well as legacy systems. Even within the system being monitored, information may be available in different formats (e.g., numerical, text, image). It is also worth noting that information about different quantities may be available at different levels of fidelity and resolution. An important challenge in data analytics for PHM is information integration, i.e., fusion of heterogeneous information available from multiple sources and activities.

Health monitoring systems have used either data-driven techniques or model-based techniques for diagnosis and prognosis. An effective framework for health diagnosis and prognosis of aging structures (subjected to physical, chemical, environmental, and mechanical degradation) needs to make use of all the available information through damage modeling, monitoring, data analytics, and uncertainty quantification techniques. This paper suggests a dynamic Bayesian network (DBN) approach for information integration, data analytics and uncertainty quantification in diagnosis and prognosis. The Bayesian network approach enables both the forward problem (uncertainty integration) and the inverse problem (risk management, resource allocation). Methods have recently been developed to

Sankaran Mahadevan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

integrate various sources of uncertainty (natural variability, data uncertainty and model uncertainty) in order to quantify the overall uncertainty in health monitoring outcome. Such methods need to be quantitatively linked to decisions regarding appropriate risk management actions through the use of structural reliability theory (Naus, 2009).

A particular problem of current interest to the authors is the application of the above concepts to the monitoring, diagnosis, prognosis, and health management of concrete structures. Concrete structures are affected by a variety of chemical, physical and mechanical degradation mechanisms such as chloride penetration, sulfate attack, carbonation, freeze-thaw cycles, shrinkage, and mechanical loading. Each of the four elements mentioned earlier – damage modeling, monitoring, data analytics and uncertainty quantification – is a difficult challenge for a heterogeneous material such as concrete. This paper outlines research needs and possible directions through a few illustrative damage modeling and health monitoring techniques for concrete structures.

2. DAMAGE MODELING

The deterioration processes in concrete structures can be classified briefly into three main groups, i.e. physical processes, chemical processes and mechanical processes (Mehta and Monteiro 2001). Sources of physical deterioration may include temperature variation and the associated thermal expansion/contraction, relative humidity variation and the associated drying shrinkage/wetting expansion, freezing and thawing cycles (i.e. frost attack), wear and abrasion etc. Sources of chemical deterioration include corrosion of reinforcement embedded in concrete, chloride penetration, carbonation, leaching of concrete constituents, acid attack, sulfate attack, and alkali-aggregate reaction etc. And sources of mechanical deterioration include externally applied overload or impact, cyclic fatigue loads, differential settlement of foundation, and seismic activity. All these sources of deterioration can alter the porosity and permeability of concrete, cause or aggravate various material flaws (such as scaling and spalling, swelling and debonding, cracking and disintegration), impair the integrity and tightness of concrete structure, and lower the loading capacity of structural member.

The physical and chemical deterioration processes of reinforced concrete structures are closely interconnected and synergistic; distinguishing any single deterioration process from the joint impact is difficult. The complexity of the aforementioned classification of deterioration processes has led the technical community to model deterioration mechanisms of concrete individually. Individual deterioration processes have been studied extensively, and significant strides have been made in developing computational models. A major current challenge is how to develop an integrated computational methodology to quantitatively assess the durability of reinforced concrete structures subjected to a variety of coupled deterioration processes that are acting simultaneously. A related issue is that damage under different deterioration processes accumulates at different rates; thus multi-physics degradation analysis also needs to account for different time scales in different processes.

In the case of concrete degradation under coupled physical/chemical processes, governing differential equations that characterize the mass/energy balance and thermodynamic/chemical equilibrium of coupled heat conduction, ionic diffusion, moisture transport and chemical reaction have been developed. A variety of multi-scale methods and continuum finite element/difference methods have been utilized to solve the interactive and nonlinear governing equations. Methods have also been pursued to connect chemical reaction products to the mechanical response of concrete (e.g., stress, displacement, crack density). The accelerating effects of cracking on the transport processes of various aggressive agents have also been considered.

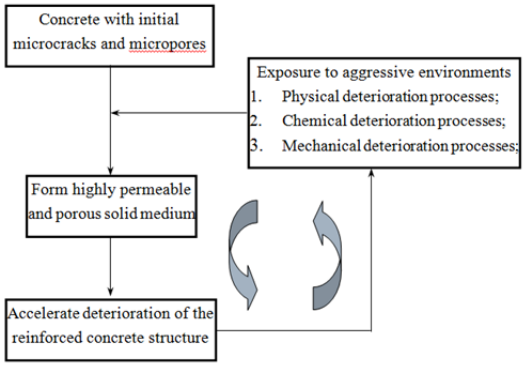


Figure 1. Multi-physics degradation of concrete

Prior to experiencing any deterioration, ordinary concrete usually possesses high porosity and low permeability. The overall connectivity of the micropore network, instead of the porosity of concrete, controls the transport properties of concrete. In other words, only interconnected micropores and microcracks in concrete contribute to the permeability of concrete and its vulnerability to deterioration. Under degrading environments, initially discontinuous micropores and microcracks grow, coalesce and finally form an interconnected network of multi-scale pores and cracks. As a result, the permeability of concrete increases, thus further accelerating the deterioration processes of the concrete structure, as shown in Fig. 1 (Chen, 2008).

Thoft-Christensen (2003) classified various deterioration models of concrete structures into three levels. Level 1 models are empirical models, which are established on the basis of direct observations on existing structural elements and do not consider the deterioration mechanism. Level 1

models have been adopted extensively in current design codes as a means of producing a rough estimate of the durability level of existing concrete structures. Level 2 models are medium level models from a sophistication viewpoint; these are based on semi-empirical or average "material parameters" (e.g., concrete permeability) and average "loading parameters" (e.g., average chloride content applied on the surface of concrete). Deterioration mechanisms are assumed to follow some formulated physical principles like Fick's law. Level 2 models have usually limited their scope to individual deterioration mechanisms. Level 3 is the most advanced level, where the modeling of the deterioration profile is based on fundamental physical, chemical and mechanical principles. Detailed information on concrete microstructure and applied environmental loading is required, and multiple coupled deterioration processes are taken into account.

A few examples of multi-physics degradation modeling, namely carbonation and chloride penetration (Level 2), and sulfate attack (Level 3), are described next for the sake of illustration.

Carbonation

Unlike physical deterioration processes such as the heat transfer and moisture transport, carbonation of concrete is essentially a chemical process. As the hydration product of Portland cement, calcium hydroxide in concrete may react with carbon dioxide dissolved in pore solution, neutralize its high alkalinity environment, and finally result in depassivation of the passive layer and initiation of reinforcement corrosion — one of the major deterioration mechanisms for reinforced concrete structures. On the other hand, as the main product of the carbonation reaction, calcium carbonate will not dissolve in water but precipitate in the pores of concrete, thus decreasing the porosity of concrete and altering its microstructure. In this case, carbonation reaction may be favorable to maintain the durability of plain concrete. Thus carbonation has opposing effects on different constituents of the material.

Based on an assumption that the carbonation front advances after the alkaline material (i.e., calcium hydroxide) has been neutralized completely, the carbonation process is dominated by the diffusion of carbon dioxide through the porous microstructure of concrete, where the concentration gradient of carbon dioxide acts as a driving force. As a neutralization reaction, the carbonation process generates a specific amount of moisture, which may affect the temporal and spatial distribution of moisture content in concrete and should be considered in the simulation of previous moisture transport process. To develop a numerical model for carbonation, several coupled processes, namely the diffusion of carbon dioxide, moisture transport, heat transfer, formation of calcium carbonate, availability of calcium hydroxide in the pore solution etc., need to be considered. A popular approach is the multifactor equation, where the diffusivity of CO_2 is assumed to be dependent on the pore relative humidity, temperature and the carbonation-induced reduction of porosity as

$$D_c = D_{c,0} \cdot F_1^*(h) \cdot F_2(T) \cdot F_3(\xi) \tag{1}$$

where F_1, F_2 and F_3 represent the effects of humidity, temperature and carbonation, respectively. Refer Saetta et al (1995) for details of models for F_1, F_2 and F_3. Saetta et al. (2004) also proposed a similar numerical model for the carbonation reaction rate as

$$\upsilon_r = \upsilon_0 \cdot f_T \cdot f_h \cdot f_c \cdot f_R \tag{2}$$

where υ_0 indicates an ideal carbonation rate at which the carbonation reaction takes place in specified ideal conditions, and f_T, f_h, f_c, and f_R represent the influences of temperature, relative humidity, concentration of free CO_2, and degree of carbonation respectively, on the reaction rate.

Chloride Penetration

Chloride-induced reinforcement corrosion is one of the major deterioration mechanisms for reinforced concrete structures exposed to marine environment, deicing salts or underground environment. It leads to a series of structural degradations, such as loss of the concrete-steel interface bond, reduction of the cross-section area of reinforcement, and cracking and spalling of the concrete cover, thus severely reducing the load carrying capacity of the structure. Considering its unique significance, substantial studies have been carried out on the chloride-induced reinforcement corrosion process for several decades.

Based on Fick's second law, the governing equation of chloride penetration in concrete is expressed as:

$$\frac{\partial C_{cl}(x,t)}{\partial t} = D_{cl} \cdot \frac{\partial^2 C_{cl}(x,t)}{\partial x^2} \tag{3}$$

where $C_{cl}(x,t)$ is the chloride content at spatial coordinate x and time t, and D_{cl} is chloride diffusivity. Chen and Mahadevan (2008) proposed the modeling of chloride-induced deterioration through a multifactor equation as

$$D_{cl} = D_{cl,0} \cdot F_2(t) \cdot F_3(C_{cl,f}) \cdot F_4(T) \cdot F_5(\rho_{local}) \tag{4}$$

where $D_{cl,0}$ is the reference or nominal chloride diffusivity when all influencing factors assume values of unity. F_2 denotes the influence of the age of concrete, which reflects the cement hydration-induced reduction in the concrete porosity with time t. F_3 represents the influence of the free chloride content $C_{cl,f}$, which reflects the hindering effect of high chloride content on the chloride diffusion. F_4 indicates the influence of temperature T, which reflects the

thermodynamic effect of high temperature on the chloride diffusion. F_5 reflects the influence of local relative crack density ρ_{local}. Chen and Mahadevan (2008) implemented this approach through a finite element-based computational methodology to link the diffusivity change to structural degradation expressed by the local relative crack density.

The above two modeling approaches use semi-empirical multifactor equations, whose parameters are calibrated using experimental data. These are Level 2 approaches using averaged parameters. An example of a Level 3 approach based on multi-scale modeling is illustrated below for sulfate attack.

Sulfate attack

When sulfate ions diffuse through a cementitious structure, they react with the cement hydration products to form expansive products. This induces strain leading to cracking and eventual failure. Sarkar (2010) developed a probabilistic computational model of concrete durability under sulfate attack that considers three processes – diffusion of ions, chemical reactions and mechanical damage accumulation due to cracking. The three processes were modelled through basic differential equations, chemical reactions and mechanics models respectively, based on continuum first principles.

There are several inputs and model parameters in the three parts of the model. Sarkar et al (2012) pursued a hierarchical Bayesian calibration approach where the parameters of each model component were calibrated using tests that progressively added the processes (i.e., first chemical alone, then chemical and diffusion, then all three). In the geochemical speciation modeling, many mineral sets are possible; their relative proportions were calibrated using experimental data.

The effect of chemical reaction products on mechanical properties such as elastic modulus and strength was computed through multi-scale modeling. Four scales were considered for homogenization and calculation of macro-level structural properties and strength degradation. These were: calcium silicate hydrate (CSH), cement paste, cement mortar, and concrete. The macro-level crack density was then connected to effective elastic modulus and diffusivity.

In summary, the above examples of concrete deterioration modeling show attempts at modeling the interactions among multiple chemical, physical and mechanical processes that operate simultaneously across multiple spatial and temporal scales. This presents unique challenges for concrete structures health monitoring. Sensing of physical, chemical and mechanical quantities is one challenge. In addition, since multiple processes are interacting in a coupled manner, it is difficult to link any observed damage to a particular deterioration process or to estimate the proportion of damage contributed by different processes.

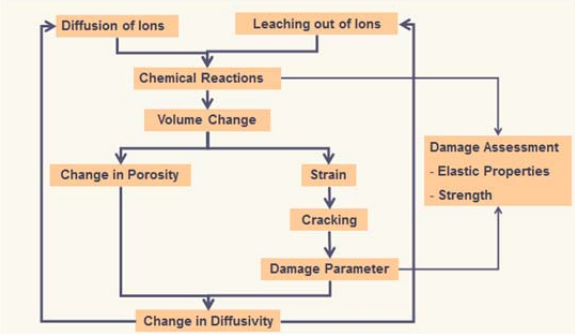


Figure 2. Multi-physics modeling of sulfate attack

3. HEALTH MONITORING

A variety of non-destructive evaluation (NDE) techniques have been studied for concrete structures. While some studies have investigated embedded sensors in concrete, we restrict this discussion to external sensing considering that the structures are already built. In a recent study led by the Oak Ridge National Laboratory, five NDE techniques were assessed for damage detection in concrete, namely shear-wave ultrasound, ground penetrating radar, impact echo, ultrasonic surface wave, and ultrasonic tomography (Clayton 2014). The techniques were compared in terms of ease of use, time consumption, and defect detection capability, and different techniques showed different advantages and disadvantages. For example, ultrasonic tomography appeared to have the best detection especially at larger depths under the surface, but was very time consuming. The first two (shear-wave ultrasound and ground penetrating radar) were found to have above average performance but some disadvantages as well.

For larger structures (e.g., containment structure in a nuclear power plant), the use of full-field imaging techniques appear promising. Some of these techniques are briefly discussed below (infrared imaging, digital image correlation, and velocimetry).

By using infrared imaging, it is possible to identify the thermal load path in a material. By tracking this thermal signature longitudinally in time, the onset of changes in the load path and hence changes in the composition of a material as well as mechanical damage in the material can be identified. Infrared imaging can also be combined with excitation techniques such as standoff acoustic sound pressure. By insonifying a material with an acoustic source, full-field vibro-thermography measurements can be made to characterize changes in the material over time. Such a methodology falls into the class of active structural health monitoring sensing methods (Mares et al, 2013).

A second approach to structural health monitoring for full-field infrared imaging is to measure the thermal response under an applied uniform heat flux. By analyzing thermal gradients in the material, regions of non-uniform material

composition such as due to the formation of defects can be identified and tracked (Sharp et al, 2014)..

Digital image correlation (DIC) has also been studied in recent years as a full-field structural health monitoring imaging technique. For example, DIC has been used to detect micro cracking in chopped fiberglass compression molded parts. The resulting image shows the principal strains in a region where a crack has formed. The strain field indicates the strains that occur under an applied static load. This method can also be used to detect localized residual strains (and stresses) after an applied load is removed. Furthermore, the method is applicable to tracking the strain that occurs under temperature or other types of environmental loading (wind, solar, etc.).

Velocimetry has also been studied as a full-field structural health monitoring imaging technique to detect subsurface nonlinearity due to material damage. For example, full-field velocimetry has been applied to monitor the ambient vibration of composite structures and data has been analyzed to detect subsurface damage in such materials. Damage indices quantify the degree of nonlinear stiffness/damping behavior that is observed locally at each measurement point in the grid. Using modern scanning laser technology, it is possible to perform these measurements for in-plane an out-of-plane vibration fields to achieve greater sensitivity to defects in composite structures. Using this technique, it has been demonstrated that the nonlinear dynamic behavior of heterogeneous materials such as the fiberglass sandwich material are indicative of subsurface damage, and that a higher frequency vibration provides for enhanced localization of the damage (Bond et al, 2013).

The aforementioned full-field measurement techniques have been applied to metallic and composite material structures. Their suitability for concrete structures is yet to be investigated. Full-field measurements also need to be supplemented by appropriate NDE and laboratory testing activities.

4. DATA ANALYTICS

Data analytics is a crucial step in processing the collected data and assembling the evidence for diagnosis and prognosis. A variety of data processing techniques have been developed during the past decades to analyze the data generated by the sensor systems. In general, health monitoring systems and sensors generate a large amount of data. For online monitoring, the amount of information grows very large, and this becomes a big data problem. A big data problem is characterized by volume, velocity and variety (heterogeneity) of data. When full-field imaging techniques are used, data analytics is challenged by the presence of heterogeneous data (numerical, text and image). The data becomes too large and complex to be stored, managed and processed by traditional database management techniques.

In recent years, several software frameworks for storage, management and retrieval of big data have been developed. The well-known Hadoop distributed file system for storing large amounts of data is scalable and fault-tolerant. MapReduce is a parallel processing framework for large-scale data processing. It consists of two segments -- Map function, where the task is subdivided and assigned to slave nodes, and Reduce function, where the results from slave nodes are aggregated to obtain final result (Prajapati, 2013).

Big data presents many issues such as data quality, relevance, re-use, decision support etc. In particular, uncertainty of inference due to data quality, and incompleteness need to be addressed. Sensitivity analysis leads to identifying the relevance of various data components, and helps to focus attention and collection efforts to the most relevant data. Additional challenges relate to data scrubbing and robust data management, as also the requirements for increased memory, storage and computing power.

Dimension reduction and data reduction are common steps in processing big data. Dimension reduction is achieved through feature selection and extraction. Two types of approaches are available for feature selection – filter approach and wrapper approach. In the wrapper approach, all possible subsets to predict the output variable are created, and the subset of variables, whose corresponding classification algorithm performs the best, is selected. In the filter approach, ranks are assigned to individual variables, and depending upon the accuracy required, the subset of variables is selected. In general, filter methods tend to be faster. In Feature Extraction, all the variables are mapped to a lower-dimensional space and models are constructed in this low-dimensional space. Principal components analysis (PCA) and factor analysis are well-known techniques that aid dimension reduction.

Prominent data reduction techniques include classification and clustering. Several different classification techniques such as decision trees, nearest neighbor classifier, neural networks and support vector machines are available. However, many of these are deterministic classifiers, whereas the Bayesian network is an uncertainty-based classifier where the available evidence is assigned to different classes with a quantified probability measure. Clustering can be either hierarchical or based on partition of the problem domain. Several different clustering techniques, such as k-means, DBSCAN, expectation maximization are available, and these need to be investigated for suitability in the present problem. For larger data sets, dimension reduction is possible through feature extraction and feature selection, in order to develop a low-dimensional representation of the available data.

After preprocessing and reducing the available data, the next step is PHM model building by learning the interrelationships. While doing this, it is advisable to use the data in a systematic manner that maximizes the information gain. An adaptive selection of data sources can be pursued, based on information-theoretic metrics. Various possible data sources are ranked based on the information gain potential and selected to train the model in decreasing order of information gain.

In summary, data about different physical quantities being measured is available in heterogeneous formats and fidelity, from multiple sources (e.g., test data, expert opinion, operational data, legacy system data, and model-based simulations). Data may be sparse about some quantities, while it may be abundant for other quantities. A systematic and rigorous approach is needed for data analytics that makes use of all available heterogeneous information. One promising approach is to use the Bayesian network (BN) machine learning approach as the organizing principle for connecting data in multiple different formats. The Bayesian network (discussed in the next section) allows the integration of various types of information that (a) occur at different times, and (b) combine in different ways (linear, nonlinear, coupled, nested, and iterative).

5. UNCERTAINTY QUANTIFICATION

Uncertainty sources in various components of the PHM model may broadly be classified into three categories: natural variability in the system properties and operating environments (aleatory uncertainty), information uncertainty due to inadequate, qualitative, missing, or erroneous data (epistemic uncertainty), and modeling uncertainty induced by assumptions and approximations (epistemic uncertainty). Much previous work has focused on variability, but a systematic approach to include data and model uncertainty sources within PHM still awaits development.

Data Uncertainty: On the one hand, sensor information may be inadequate, due to sparse, imprecise, qualitative, subjective, faulty, or missing data. On the other hand, one may be confronted with a large volume of heterogeneous data (big data), involving significant uncertainty in data quality, relevance, and data processing. In the context of a probabilistic framework, both situations lead to uncertainty in the distribution parameters and distribution types of the variables being studied, and the Bayesian approach is naturally suited to handle such data cases and update the description with new information. Flexible parametric or non-parametric representations can be developed within the Bayesian framework to handle such epistemic uncertainty (Sankararaman and Mahadevan, 2011). An important recent development is the extension of global sensitivity analysis to quantify and distinguish the relative contributions of aleatory uncertainty vs. epistemic uncertainty (Sankararaman and Mahadevan, 2013a).

Model Uncertainty: The challenges in developing a computational framework for concrete degradation modeling that mathematically represents the interactions among the multi-physics degradation processes and their relation to the quantities being measured by sensors were discussed earlier. The models for various processes could be based on first principles or regression of empirical data. For some components there may not even be any mathematical models available, but perhaps reliability data from past experience or literature. The Bayesian network offers a systematic approach to integrate such heterogeneous information. Quantification of the model uncertainty resulting from such heterogeneous information could be studied w.r.t. three categories, namely, model parameters, model form, and solution approximations; and the corresponding activities to quantify them are calibration, validation and verification, respectively. Model parameters are estimated using calibration data, and Bayesian calibration constructs probability distributions for the model parameters. Model form uncertainty may be quantified in two ways: either through a validation metric, based on validation data, or as model form error (also referred to as model discrepancy or model inadequacy). Model form error can be estimated along with the model parameters using calibration and/or validation data, based on the comparison of model prediction against physical observation, and after accounting for solution approximation errors, uncertainty quantification errors, and measurement errors in the inputs and outputs (Liang and Mahadevan, 2011)

Probabilistic graphical models for machine learning such as Bayesian networks (Jensen, 1996) have shown much effectiveness in the integration of information across multiple components and physics in several application domains. Dynamic Bayesian networks (DBNs) have been used for systems evolving in time, and recent work has extended DBNs to include heterogeneous information in diagnosis and prognosis (Bartram and Mahadevan, 2014). The Bayesian network is able to include asynchronous information from different sources. Also, Bayesian networks can be built in a hierarchical manner, by composing component-level networks to form a system-level network.

In summary, data and model uncertainty sources need to be systematically included in the PHM of concrete structures, and the Bayesian network offers such a systematic and comprehensive approach for the aggregation of uncertainty from multiple sources and heterogeneous information. The Bayesian network facilitates both forward propagation of uncertainty and the inverse problem of decision-making (e.g., sensor layout design) in order to achieve uncertainty reduction. The Bayesian approach has been used to quantify the uncertainty in each step of diagnosis and prognosis (Sankararaman et al, 2011; Sankararaman and Mahadevan, 2013b). Connection of these uncertainty quantification techniques to risk assessment and risk management

decisions through the use of structural reliability theory needs to be investigated (Naus, 2009).

6. CONCLUSION

This paper discussed challenges encountered in four elements of PHM for concrete structures – degradation modeling, sensor measurement, data analytics and uncertainty quantification. Illustrative techniques and ongoing challenges in each direction were briefly discussed. An important current need is the development of an effective framework for PHM of concrete structures that combines the state-of-the-art techniques in each of the four elements, overcomes challenges such as feasibility, complexity and scalability, and develops confidence in PHM result. Such a comprehensive approach will facilitate the development of a quantitative, risk-informed framework for structural health management.

ACKNOWLEDGEMENT

This study is partly funded by Battelle Energy Alliance LLC at Idaho National Laboratory (Contract No. 0014530, Monitors: Bruce Hallbert, Vivek Agarwal). The support is gratefully acknowledged.

REFERENCES

Naus, D.J (2009), "The Management of Aging in Nuclear Power Plant Concrete Structures," Journal of the Minerals, Metals and Materials Society

Mehta, P.K. and Monteiro, P. (2001). Concrete — Microstructure, Properties and Materials, Prentice-Hall, Englewood Cliffs, New Jersey.

Chen, D.(2006), "Computational Framework for Durability Assessment of Reinforced Concrete Structures under Coupled Deterioration Processes," Ph.D. Dissertation, Vanderbilt University, Nashville, TN.

Thoft-Christensen, P. (2003) Corrosion and Cracking of Reinforced Concrete, in Life-Cycle Performance of Deteriorating Structures: Assessment, Design and Management, ASCE, pp. 26-36.

Saetta, A.V. and Vitaliani, R.V. (2004) Experimental investigation and numerical modeling of carbonation process in reinforced concrete structures, Cement and Concrete Research, Vol. 34, No. 4, pp. 571-579.

Chen, D., and Mahadevan, S. (2008), "Chloride-Induced Reinforcement Corrosion and Concrete Cracking Simulation," Cement and Concrete Composites, Vol. 30, pp. 227-238, No. 3.

Sarkar, S., Mahadevan, S., J.C.L. Meeussen, H. van der Sloot, D.S. Kosson (2010), "Numerical Simulation of Cementitious Materials Degradation Under External Sulfate Attack," Cement and Concrete Composites, Vol. 32, No. 3, pp. 241-252.

Sarkar, S., Kosson, D.S., Mahadevan, S., Meeussen, J.C.L., van der Sloot, H., Arnold, J.R., Brown, K.G (2012). " Bayesian calibration of thermodynamic parameters for geochemical speciation modeling of cementitious materials," Cement and Concrete Research, Vol. 42, No. 7, pp. 889-902.

Clayton, D. (2014), "Nondestructive Evaluation Techniques for Nuclear Power Plant Concrete Structures," Light Water Reactor Sustainability (LWRS) Newsletter, Issue 14, U. S. Department of Energy.

Mares, J., Miller, J., Rhoads, J., Son, S., Groven, L., Sharp, N., and Adams, D. (2013), "Thermal and Mechanical Response of PBX 9501, PBS 9501, and 900-21 under High-Frequency Mechanical Excitation,", Journal of Applied Physics, 113, 084904.

Sharp, N., P. O'Regan, Adams, D. E., Caruthers, J., David, A., and Suchomel, M. (2014), "Lithium-Ion Battery Electrode Inspection using Pulse Thermography,"NDT&E International, Vol. 64, pp. 41-51.

Bond, R., Underwood, S., Adams, D., and Cummins, J. (2013), "Structural Health Monitoring-Based Methodologies for Managing Uncertainty in Aircraft Structural Life Assessment," Proceedings of the 9th International Workshop on Structural Health Monitoring, Palo Alto, CA.

Prajapati, V.(2013), Big Data Analytics with R and Hadoop, Packt Publishing, Birmingham, UK.

Sankararaman, S., and Mahadevan, S. (2011), " Likelihood-Based Representation of Epistemic Uncertainty due to Sparse Point Data and Interval Data," Reliability Engineering and System Safety. Vol. 96, No. 7, pp. 814-824.

Sankararaman, S., and Mahadevan, S. (2013), "Separating the Contributions of Variability and Parameter Uncertainty in Probability Distributions," Reliability Engineering & System Safety, Vol. 112, pp. 187-199.

Liang, B., and Mahadevan, S. (2011), " Error and Uncertainty Quantification and Sensitivity Analysis of Mechanics Computational Models," International Journal for Uncertainty Quantification. Vol. 1, No. 2, pp. 147-161.

Jensen, F. V., An Introduction to Bayesian Networks. Springer-Verlag, 1996.

Bartram, G., and Mahadevan, S. (2014), "Integration of Heterogeneous Information in SHM Models," Structural Control and Health Monitoring, Vol. 21, No. 3, pp. 403-422.

Sankararaman, S., Ling, Y., Shantz, C. and Mahadevan, S. (2011), "Uncertainty Quantification in Fatigue Crack Growth Prognosis," International Journal of Prognostics and Health Management, Vol. 2, no. 1.

Sankararaman, S., and Mahadevan, S.(2013), "Bayesian methodology for diagnosis uncertainty quantification and health monitoring," Structural Control and Health Monitoring, Vol. 20, pp. 88-106.

Investigating Vibration Properties of a Planetary Gear Set with a Cracked Tooth in a Planet Gear

Xihui Liang [1] and Ming J. Zuo [2]

[1,2]*Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, T6G2G8, Canada*
ming.zuo@ualberta.ca

ABSTRACT

Comparing with fixed shaft gearbox, vibration properties of planetary gearbox are much more complicated. In a planetary gearbox, there are multiple vibration sources as several pairs of sun-planet gears and several pairs of ring-planet gears mesh simultaneously. In addition, the signal transmission path changes due to the rotation of the carrier. To facilitate fault detection of a planetary gearbox and avoid catastrophic consequences caused by gear failures, it is essential to understand the vibration properties of a planetary gearbox. This paper aims to simulate vibration signals and investigate vibration properties of a planetary gear set when there is a cracked tooth in a planet gear. Displacement signals of the sun gear and the planet gear, and resultant acceleration signals of the whole planetary gear set will be simulated and investigated. Previous work mainly focuses on the vibration properties of a single component, like the sun gear, the planet gear or the carrier. This paper simulated the vibration signal of a whole planetary gear set when there is a cracked tooth in a planet gear. In addition, fault symptoms will be revealed, which can be utilized to detect the crack in the planet gear. Finally, the proposed approach is experimentally validated.

1. INTRODUCTION

Planetary gears are widely used in aeronautic and industrial applications because of properties of compactness and high torque-to-weight ratio. A planetary gear set consists normally of a centrally pivoted sun gear, a ring gear and several planet gears that mesh with the sun gear and the ring gear simultaneously as shown in Figure 1.

The vibration signals of a planetary gearbox are more complicated comparing with that of a fixed-shaft gearbox. In a planetary gearbox, there are multiple vibration sources as several pairs of sun-planet gears and several pairs of ring-planet gears mesh simultaneously. In addition, signal transmission path changes due to the rotation of the carrier. Multiple vibration sources and the effect of transmission path lead to complexity of fault detection (Liang, Zuo and Hoseini, 2014).

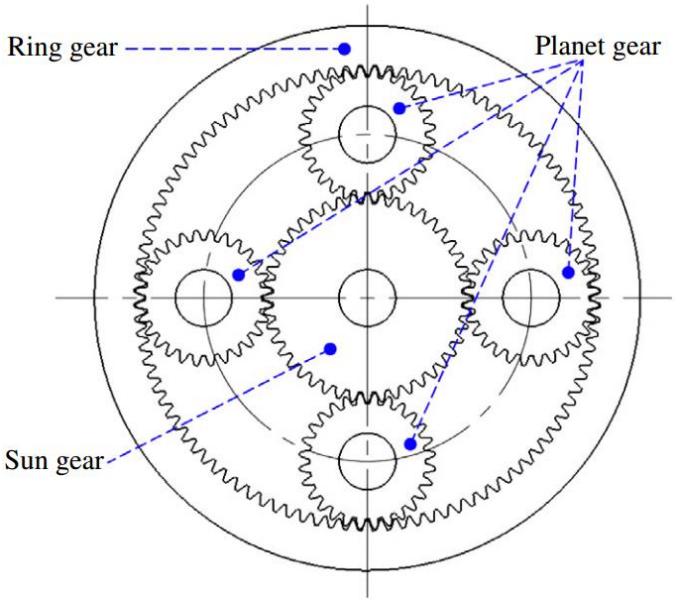


Figure 1. A planetary gear set having four planet gears (Lei, Lin, Zuo and He, 2014)

A few studies investigated vibration properties of the planetary gearbox. Zhang, Khawaja, Patrick, et al. (2008) applied the blind deconvolution algorithms to denoise the vibration signals collected from a testbed of the helicopter main gearbox subjected to a seeded fault. Inalpolat and Kahraman (2009) proposed a simplified mathematical model to describe the mechanisms leading to modulation sidebands of planetary gear sets. Inalpolat and Kahraman (2010) predicted modulation sidebands of a planetary gear set having manufacturing errors. Chen, Vachtsevanos and Orchard (2012) proposed an integrated remaining useful life prediction method which was validated by successfully applying the method to a seeded fault test for a UH-60 helicopter planetary gear plate. Feng and Zuo (2012)

Xihui Liang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

mathematically modeled tooth pitting and tooth wear by applying amplitude modulation and frequency modulation, and then analyzed the spectral structure of the vibration signals of a planetary gear set. Patrick, Ferri and Vachtsevanos (2012) studied the effect of planetary gear carrier-plate cracks on vibration spectrum. Liang, Zuo and Hoseini (2014) investigated the vibration properties of a planetary gear set when there is a cracked tooth in the sun gear. Chen and Shao (2011) studied the dynamic features of a planetary gear set with tooth crack under different sizes and inclination angles. The displacement signals of the sun gear and the planet gear were investigated when a tooth crack was present on the sun gear or the planet gear. However, the effect of transmission path was not considered in their analysis. They did not model the resultant vibration signals of the whole gearbox. In practical applications, sensors are commonly mounted on the housing of the gearbox to capture the vibration signals. The signals acquired by sensors are the resultant vibration signals of the whole gearbox. They are not the vibration signals of the sun gear or a single planet gear. In this study, the resultant vibration signals of a planetary gear set will be modeled and then analyzed when there is a cracked tooth in the planet gear.

2. MODELING OF VIBRATION SIGNALS

Liang, Zuo and Hoseini (2014) simulated and investigated the vibration signals of a planetary gear set when there is a cracked tooth in the sun gear. The method proposed by Liang, Zuo and Hoseini (2014) will be applied directly in this paper. This study does not intend to propose a new method to model the vibration signals of a planetary gear set. This paper focuses on exploring vibration properties, and then finds the fault symptoms of a planetary gear set when there is a cracked tooth in the planet gear. Two steps are required to obtain the resultant vibration signals of a planetary gear set. First of all, a dynamic model will be applied to simulate the vibration signals of each gear, including the sun gear, each planet gear and the ring gear. Then, resultant vibration signals will be modeled considering multiple vibration sources and effect of transmission path.

2.1. Dynamic Modeling of a Planetary Gear Set

The dynamic model used in this study is the same as that used by Liang, Zuo and Hoseini (2014) except for differences of sun-planet mesh stiffness. The differences of sun-planet mesh stiffness will be described in detail in Section 3. Figure 2 shows the dynamic model that will be used in this study. It is a nonlinear two-dimensional lumped-mass model. Each component has three degrees of freedom. Total, it has 9+3*N* degrees of freedom as a planet gear set has one sun gear, one ring gear, one carrier and *N* planet gears. All the coordinate systems are fixed on the carrier. Figure 2 shows locations and positions of all coordinate systems in the initial time (time zero). Equations of motion of the dynamic model will not be included in this paper. Equations can be found in Liang, Zuo and Hoseini (2014).

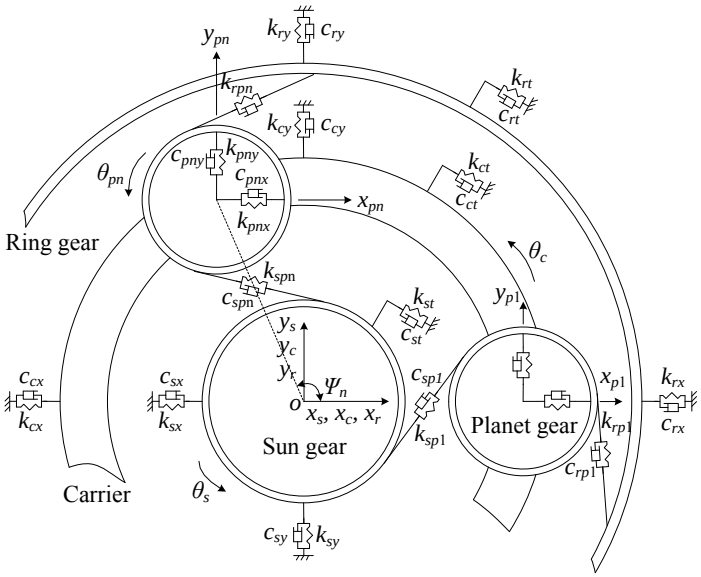


Figure 2. Dynamic modeling of a planetary gear set (Liang, Zuo and Hoseini, 2014)

2.2. Resultant Vibration Signals

A dynamic model of a planetary gear set was described in Section 2.1. Equations of motion of the dynamic model can be built correspondingly. Numerically solving the equations, vibration signals of the sun gear, the ring gear, each planet gear and the carrier can be obtained. After that, resultant vibration signal of the planetary gear set can be modeled incorporating multiple vibration sources and the effect of transmission path. The resultant vibration signal is expressed as weighted summation of acceleration of each planet gear as shown in Equation (1) (Liang, Zuo and Hoseini, 2014).

$$a(t) = \sum_{n=1}^{N} w_n(t) a_n(t) \qquad (1)$$

where $w_n(t)$ is the weighting function which counts for the effect of the transmission path; $a_n(t)$ represents acceleration of the n^{th} planet gear, which is obtained through dynamic simulation.

A Hamming function is used to model the effect of transmission path. The Hamming function assumes that as planet *n* approaches transducer location, its influence increases, reaching its maximum when planet *n* is closest to transducer location, then, its influence decreases as the planet goes away from the transducer.

$$w_i(t) = 0.54 - 0.46\cos(w_c t + \psi_n) \qquad (2)$$

where w_c is carrier angular frequency; ψ_n is phase angle corresponding to the n^{th} planet gear.

3. CRACK MODELING AND MESH STIFFNESS EVALUATION

Gear tooth crack is a common failure mode in a gear transmission system. It may occur in the sun gear, a planet gear or the ring gear. When a cracked tooth is in the sun gear, the cracked tooth will mesh with all planet gears. Therefore, mesh stiffness of all sun-planet gear pairs will be affected. While if a cracked tooth is in a planet gear, only mesh stiffness of one pair of sun-planet gear pair is affected.

Tooth crack mostly initiates at the critical area of a gear tooth root (area of the maximum principle stress), and the propagation paths are smooth, continuous, and in most cases, rather straight with only a slight curvature as shown in Figure 3 (Belsak and Flasker, 2007). Liang, Zuo and Pandey simplified the crack growth path as a straight line (the red line) starting from the critical area of the tooth root. The same model developed by Liang, Zuo and Pandey (2014) will be applied in this study.

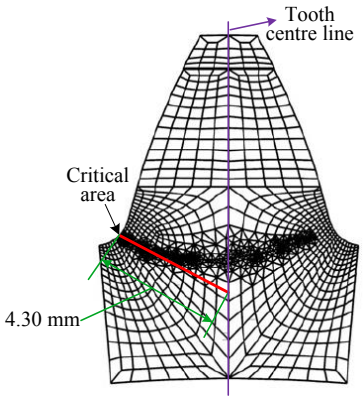


Figure 3. Crack propagation path (Belsak and Flasker, 2007)

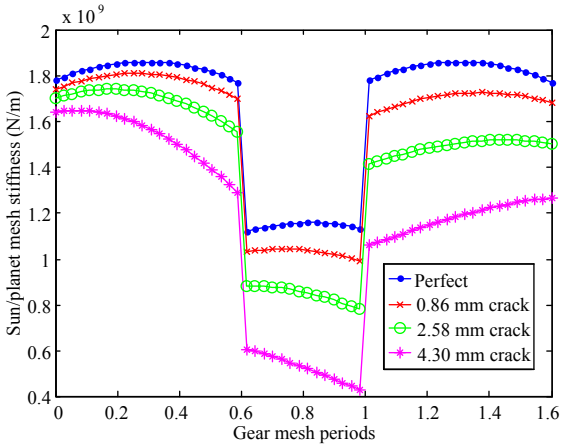


Figure 4. Mesh stiffness of a sun-planet gear pair (Liang, Zuo and Pandey, 2014)

Potential energy method used by Liang, Zuo and Pandey (2014) is applied directly in this study to evaluate the mesh stiffness of a planetary gear set in the perfect and the cracked tooth condition. Figure 4 shows the mesh stiffness of a pair of sun-planet gear when different crack levels are present on a planet gear tooth. With the growth of tooth crack, the mesh stiffness will decrease correspondingly. The reduction of mesh stiffness will cause the vibration signals behavior abnormally, which can be used to detect the tooth fault.

4. VIBRATION SIGNALS OF SUN GEAR AND PLANET GEAR

In this section, vibration signals of a planetary gear set are numerically simulated using MATLAB ode15s solver. Physical parameters of the planetary gear set are listed in Table 1. A constant torque of 450 N.m is applied to the sun gear and the rotation speed of the carrier is 8.87 r/min. Gear mesh damping is assumed to be proportional to the mesh stiffness (Tian, Zuo and Fyfe, 2004).

Table 1. Physical parameters of a planetary gear set (Liang, Zuo and Hoseini, 2014)

Parameters	Sun gear	Planet gear	Ring gear	Carrier
Number of teeth	19	31	81	---
Module (mm)	3.2	3.2	3.2	---
Pressure angle	20°	20°	20°	---
Mass (kg)	0.700	1.822	5.982	10.000
Face width (m)	0.0381	0.0381	0.0381	---
Young's modulus	2.068×10^{11}	2.068×10^{11}	2.068×10^{11}	---
Poisson's ratio	0.3	0.3	0.3	---
Base circle radius	28.3	46.2	120.8	---
Root circle radius	26.2	45.2	132.6	---
Bearing Stiffness	$k_{sx}=k_{sy}=k_{rx}=k_{ry}=k_{cx}=k_{cy}=k_{pnx}=k_{pny}=1.0\times10^{8}$ N.m			
Bearing damping	$c_{sx}=c_{sy}=c_{rx}=c_{ry}=c_{cx}=c_{cy}=c_{pnx}=c_{pny}=1.5\times10^{3}$ Ns/m			

Figure 5 presents displacement signals of the planet gear that has a cracked tooth. The planet gear has 31 teeth. In the time duration of 31 gear mesh periods, the cracked tooth will mesh one time. It is observable from Figure 5 that large amplitude (fault symptom) of the displacement signal is generated when the cracked tooth is in meshing. As the crack grows, the amplitude of the fault symptoms increases accordingly. The fault symptom will repeat every 31 gear mesh periods. In Figure 5, the fault symptom mainly appears in the y-direction displacement. Actually, the fault symptom may mainly appear in the x-direction displacement or in the y-direction displacement or evenly in the two directions, which depends on the location of the planet gear.

Since the ring gear has 81 teeth, the planet gear returns to its original position after 81 meshes. Figure 6 plots the center locus of the sun gear in 81 gear mesh periods when a planet gear tooth has different crack lengths. When the planet gear is in perfect condition, 81 spikes can be observed which corresponding to 81 gear meshes. When the planet gear has a cracked tooth, the cracked tooth will mesh two or three times (81/31) in 81 gear mesh periods. Figure 6 shows the condition when three meshes happen in 81 gear meshes. In the condition of 0.86 mm crack, three bigger spikes can be observed. Time duration of spike 1 and spike 2 is 31 gear

mesh periods. Similarly, it is 31 gear mesh periods between spike 2 and spike 3. The time duration of spike 3 and spike 1 is 19 (81-62) gear mesh periods. It is predicable that the 4th bigger spike will show after 31 gear mesh periods of spike 3. In the condition of 2.58 mm crack, the three spikes (1, 2 and 3) become even larger comparing to the condition of 0.86 mm crack. When the cracked tooth is in meshing, a spike will be generated due to the low stiffness of cracked tooth pair, like spike 1. Spike 1' is generated along with spike 1 by the reaction force induced by the bigger amplitude of spike 1. Same situation applies to spike 2' and spike 3'. Overall, clear fault symptoms show in the vibration signals of sun gear and planet gear.

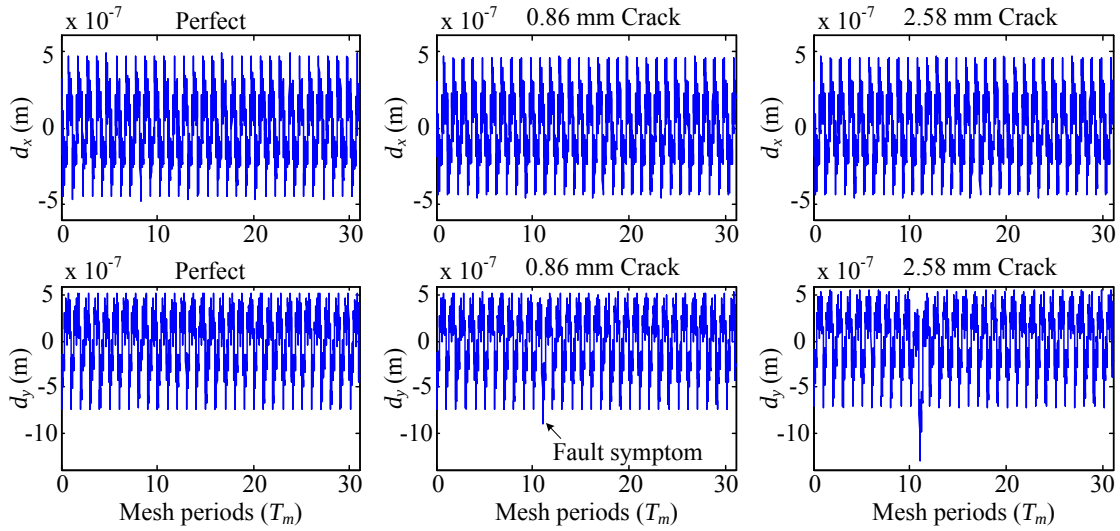


Figure 5. Displacement signals of the sun gear
d_x: displacement in x-direction; d_y: displacement in y-direction

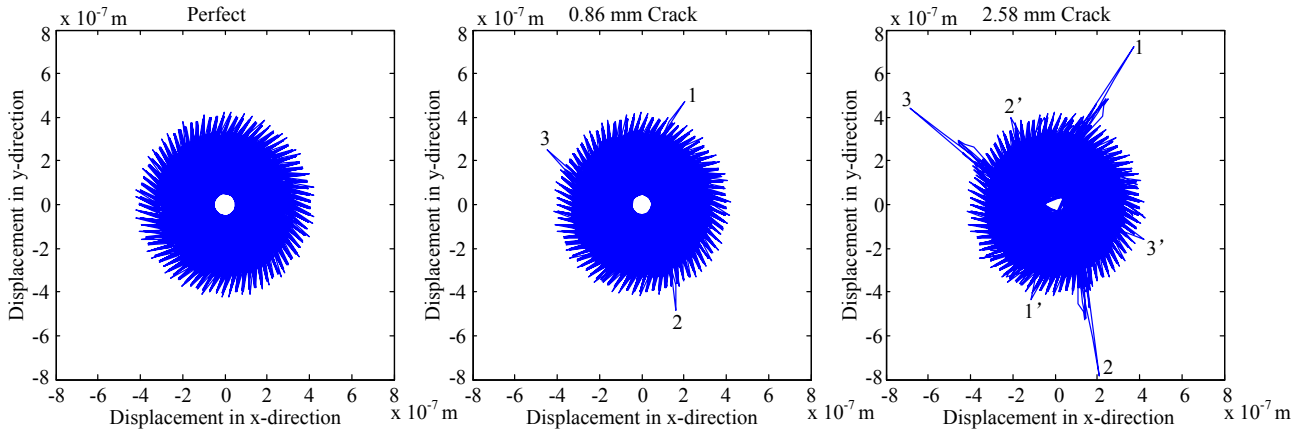


Figure 6. Center locus of the sun gear

5. RESULTANT VIBRATION SIGNALS

Applying Equation (1), resultant signal of a planetary gear set (parameters are listed in Table 1) can be generated. Figure 7 shows the resultant vibration signals in one revolution of the carrier (81 gear mesh periods). Three health conditions are plotted: perfect condition, 0.86 mm crack in one tooth of a planet gear and 2.58 mm crack in one tooth of a planet gear. The symbol a_y represents y-direction acceleration of the planetary gear set. In one revolution of the carrier, the cracked tooth should mesh three times. Three fault symptoms should appear in one revolution of the carrier. However, in the 0.86 mm crack, only one bigger spike is observed (see the red elliptical circle). In the 2.58 mm crack, two bigger spikes are observed. Therefore, some spikes are attenuated or disappeared. This is caused by the effect of transmission path. If the cracked tooth is meshing far from a transducer, the fault symptoms cannot be acquired by the transducer. Figure 8 presents frequency spectrum of simulated resultant vibration signals of a planetary gear set in different health conditions. In the perfect condition, sizable amplitudes are marked in Figure

8 and they all show in the following locations: nf_m if n is an integer and a multiple of 4, $nf_m \pm f_c$ if n is an odd integer, $nf_m \pm 2f_c$ if n is an even integer but not a multiple of 4 (Liang, Zuo and Hoseini, 2014), where f_m represents gear mesh frequency and f_c denotes rotation frequency of the carrier. When crack is present on one tooth of a planet gear, these sizable amplitudes are rarely affected. Some sidebands (see the area circled by red lines) appear due to the tooth crack even it is not obvious in Figure 8.

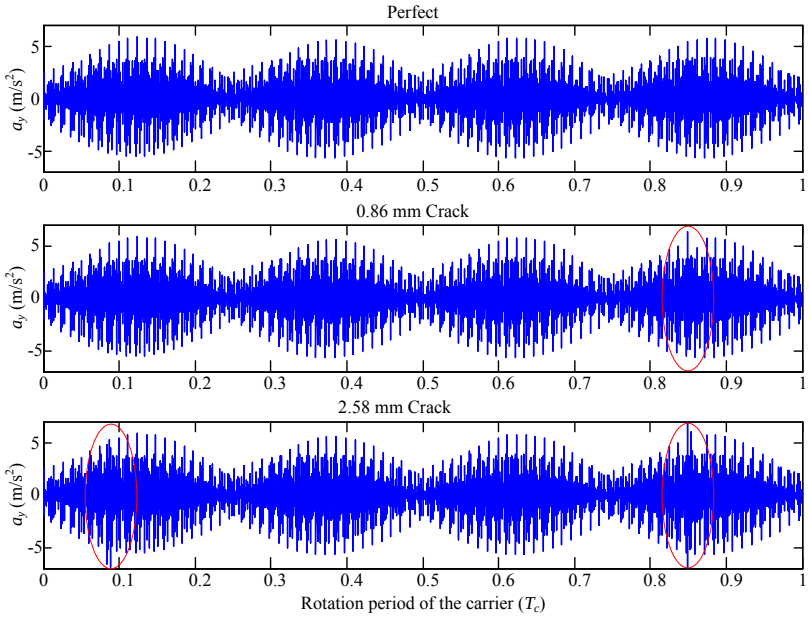


Figure 7. Simulated resultant vibration signal of a planetary gear set

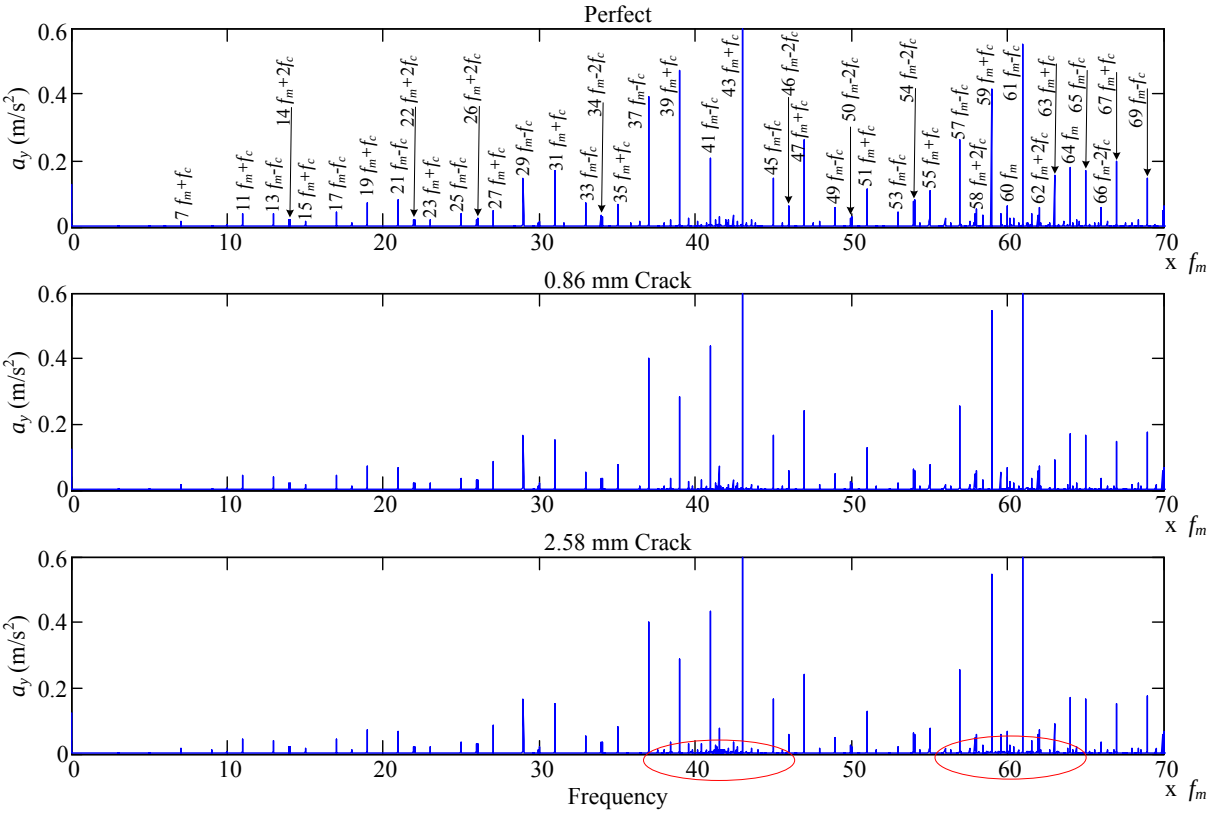


Figure 8. Frequency spectrum of simulated resultant vibration signals

Figure 9 gives zoomed-in plot of frequency region between 43 f_m and 45 f_m. Many sidebands appear but they are not symmetric. Sizable sidebands appear at $mf_m \pm nf_{planet} \pm kf_c$ or $mf_m \pm nf_{planet} \pm kf_p$, where m, n and k are all integers; f_p represents rotation frequency of the planet gear and f_{planet} denotes characteristic frequency of the faulty planet gear. For the planetary gear set we used in this studied, n and k can take the following integer values: $0 \le n \le 15$ and $0 \le k \le 1$. The characteristic frequency of the cracked planet gear can be calculated as follows (Feng et al. 2012):

$$f_{planet} = f_m / Z_p \quad (3)$$

where Z_p denotes teeth number of the planet gear.

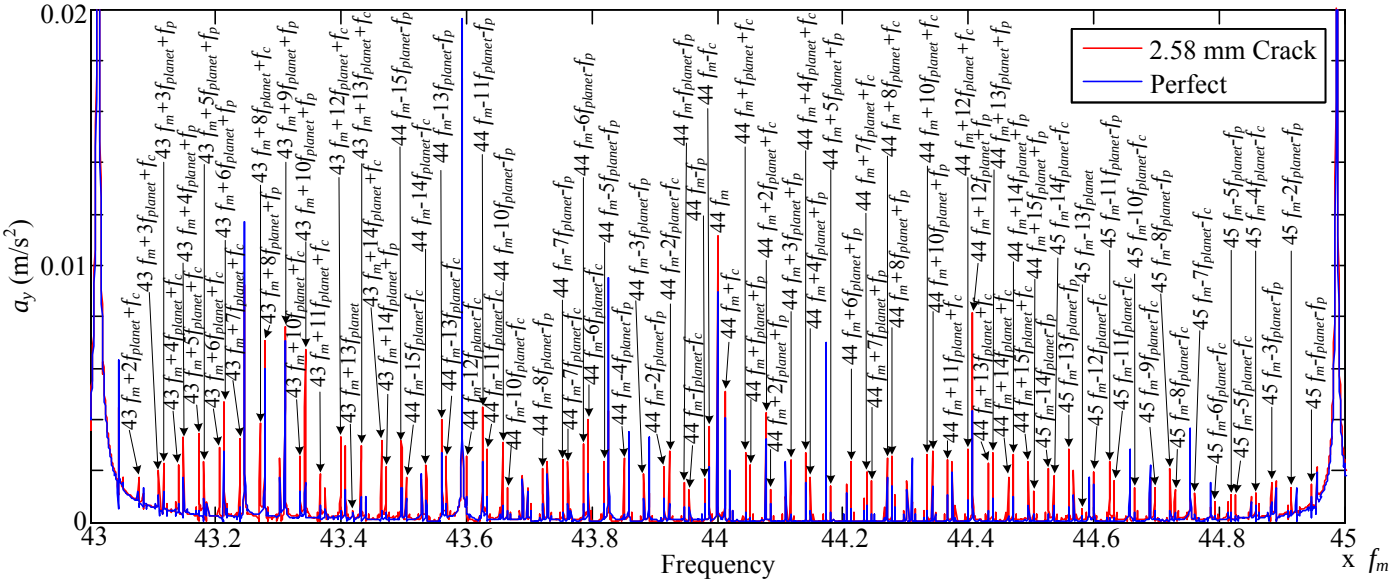


Figure 9. Zoomed-in frequency spectrum of simulated resultant vibration signals

6. EXPERIMENTAL VALIDATION

Acceleration signals are acquired from a planetary gearbox to validate simulated resultant vibration signals and fault symptoms discovered in this study. Figure 10 shows the experimental test rig whose parameters are listed in Table 2. An acceleration sensor was installed on top surface of the housing of 2nd stage planetary gearbox and vertical acceleration signals of the gearbox were recorded. The configuration and parameters of the 2nd stage planetary gear set are the same as that of the planetary gear set used for the signal simulation. The rotation speed of the carrier is 8.87 r/min that is the same carrier speed used in the simulation. When the crack length is small, fault symptoms may be submerged in the noise and hard to be detected. To amplify the fault symptoms, 4.3 mm tooth crack was created in a planet gear tooth as shown in Figure 11.

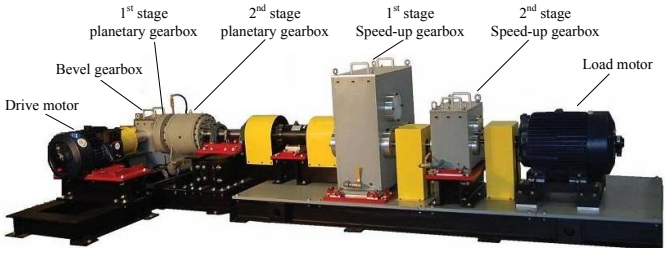


Figure 10. Experimental test rig

Table 2. Parameters of experimental test rig

Gearbox	Bevel stage		First stage planetary			Second stage		
Gear	Input	Output	Sun	Planet	Ring	Sun	Planet	Ring
No. of	18	72	28	62 (4)	152	19	31 (4)	81

Note: The number of planet gears is indicated in the parenthesis.

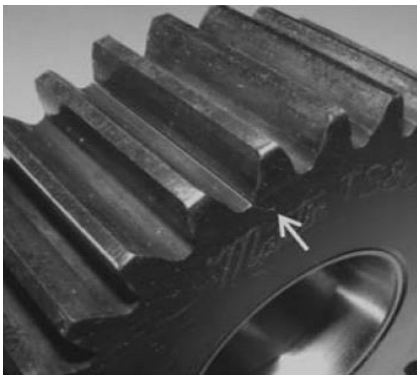


Figure 11. 4.3 mm manually made tooth crack in planet gear

Figure 12 shows the frequency spectrum of experimental vibration signals. In Figure 12, "Motor" represents rotation frequency of drive motor; "MBvl" denotes mesh frequency of bevel gears; "f_{m1}" and "f_{c1}" means mesh frequency and carrier

rotation frequency of 1st stage planetary gearbox, respectively. These frequencies are not relevant to the 2nd stage planetary gearbox. All other marked frequency components located at the following locations: nf_m if n is an integer and a multiple of 4, $nf_m \pm f_c$ if n is an odd integer, $nf_m \pm 2f_c$ if n is an even integer but not a multiple of 4.

Figure 13 describes frequency components of the experimental signal in the frequency region from 43 f_m to 45 f_m. Sidebands are not symmetric and sizable sidebands located at $mf_m \pm nf_{planet} \pm kf_c$ or $mf_m \pm nf_{planet} \pm kf_p$, where m, n and k are all integers. The sidebands locations are the same as that anticipated in Section 5.

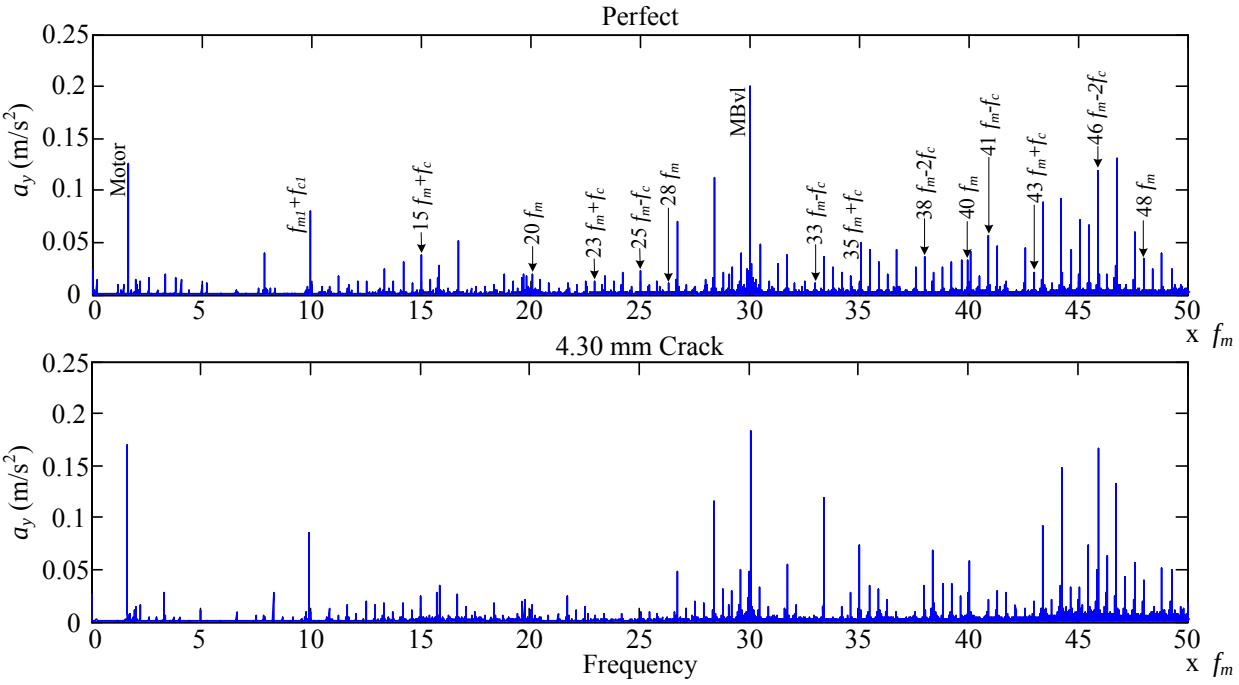


Figure 12. Frequency spectrum of experimental vibration signals

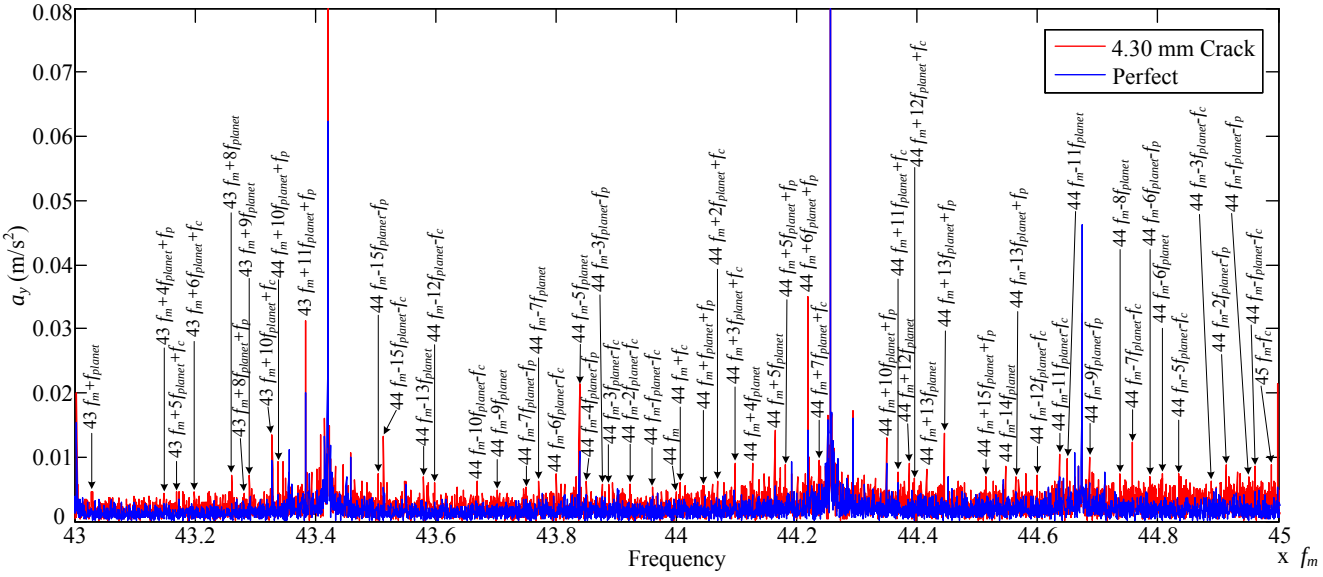


Figure 13. Zoomed-in frequency spectrum of experimental vibration signal

7. CONCLUSION

In this study, the vibration signals of a planetary gear set are simulated and investigated when there is a cracked tooth in a planet gear. When there is a cracked tooth in a planet gear, regular fault symptoms appear in the vibration signals of sun gear and planet gear. The fault symptom appears in every Z_p meshes. The fault symptoms enlarge along with the growth of crack. Some fault symptoms attenuate or disappear in the resultant vibration signal. This is due to the effect of transmission path which is caused by the rotation of carrier. Asymmetric sidebands appear when there is a cracked tooth in a planet gear. The locations of these sidebands are investigated and found, which can be used to detect tooth crack fault. Experimental validations are performed to demonstrate the correctness of the anticipated sideband locations.

ACKNOWLEDGEMENT

This research is supported by the Natural Science and Engineering Research Council of Canada (NSERC) and the China Scholarship Council (CSC).

REFERENCES

Belsak, A., & Flasker, J. (2007). Detecting cracks in the tooth root of gears. *Engineering Failure Analysis*, vol. 14, pp. 1466-1475.

Chen, Z., & Shao, Y. (2011). Dynamic simulation of spur gear with tooth root crack propagating along tooth width and crack depth. *Engineering Failure Analysis*, vol. 18, pp. 2149-2164.

Feng, Z., & Zuo, M. J. (2012). Vibration signal models for fault diagnosis of planetary gearboxes. *Journal of Sound and Vibration*, vol. 331, pp. 4919-4939.

Lei, Y., Lin, J., Zuo, M. J., & He, Z. (2014). Condition monitoring and fault diagnosis of planetary gearboxes: A review. *Measurement*, vol. 48, pp. 292-305.

Liang, X., Zuo, M.J., & Pandey, M. (2014). Analytically evaluating the influence of crack on the mesh stiffness of a planetary gear set. *Mechanism and Machine Theory*, vol. 76, pp. 20-38.

Liang, X., Zuo, M. J., & Hoseini, M. (2014). Understanding vibration properties of a planetary gear set for fault detection. *IEEE International Conference on Prognostics and Health Management*, June 22-25, Spokane, Washington, USA.

Inalpolat, M., & Kahraman, A. (2009). A theoretical and experimental investigation of modulation sidebands of planetary gear sets. *Journal of Sound and Vibration*, vol. 323, pp. 677-696.

Inalpolat, M., & Kahraman, A. (2010). A dynamic model to predict modulation sidebands of a planetary gear set having manufacturing errors. *Journal of Sound and Vibration*, vol. 329, pp. 371-393.

Patrick, R., Ferri, A., and Vachtsevanos, G. (2012). Effect of Planetary Gear Carrier-Plate Cracks on Vibration Spectrum. *Journal of Vibration and Acoustics*, vol. 134, pp. 1- 12.

Tian, X., Zuo, M. J. & Fyfe, K. (2004). Analysis of the vibration response of a gearbox with gear tooth faults. *Proceedings of ASME International Mechanical Engineering Congress and Exposition* (1-9). November 13-19, Anaheim, California, USA.

Standards for Prognostics and Health Management (PHM) Techniques within Manufacturing Operations

Gregory W. Vogl[1], Brian A. Weiss[1], and M. Alkan Donmez[1]

[1]*National Institute of Standards and Technology, Gaithersburg, Maryland, 20899, USA*

gregory.vogl@nist.gov
brian.weiss@nist.gov
alkan.donmez@nist.gov

ABSTRACT

Prognostics and health management (PHM) technologies reduce time and costs for maintenance of products or processes through efficient and cost-effective diagnostic and prognostic activities. These activities aim to provide actionable information to enable intelligent decision-making for improved performance, safety, reliability, and maintainability. Thoughtful PHM techniques can have a dramatic impact on manufacturing operations, and standards for PHM system development, data collection and analysis techniques, data management, system training, and software interoperability need to exist for manufacturing. The National Institute of Standards and Technology (NIST) conducted a survey of PHM-related standards applicable to manufacturing systems to determine the needs addressed by such standards, the extent of these standards, and any commonalities as well as potential gaps among the documents. Standards from various national and international organizations are summarized, including those from the International Electrotechnical Commission, the International Organization for Standardization, and SAE International. Finally, areas for future PHM-related standards development are identified.

1. PHM ENABLES SMART MANUFACTURING

Prognostics and health management (PHM) systems and technologies enable maintenance action on products and processes based on need, determined by the current system condition via diagnostic analyses and/or the expected future condition through prognostic methods. PHM techniques are in contrast to the use of schedules (i.e., preventative maintenance) where maintenance is conducted on specific time intervals (United States Army, 2013). PHM aims to reduce burdensome maintenance tasks while increasing the availability, safety, and cost effectiveness for the products and processes to which it is applied. In this sense, PHM enables smart manufacturing by optimizing maintenance operations via data collection, diagnostics, and prognostics as well as usage monitoring.

Gregory W. Vogl et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.1. National Strategic Needs in Manufacturing

The United States is beginning to gain ground in reestablishing its manufacturing dominance through research and development in a wide-range of advanced technologies. Additive manufacturing, robotics, data analytics, cloud computing, and intelligent maintenance are just a few evolutionary technologies that are actively being refined. These technologies can have a tremendous impact on U.S. manufacturing that would "increase productivity, efficiency and innovation, speed-to-market, and flexibility" (Ludwig & Spiegel, 2014).

The National Institute of Standards and Technology (NIST) is focused on advancing, documenting, and standardizing industry practices in many of these new technologies. Standards have a well-documented history of impact within the national and global manufacturing community (Ludwig & Spiegel, 2014). NIST has a strong history of working with industry to develop standards and guidelines to promote best practices and further manufacturing competitiveness (Bostelman, Teizer, Ray, Agronin & Albanese, 2014, Hunten, Barnard Feeney & Srinivasan, 2013, Lee, Song & Gu, 2012, Marvel & Bostelman, 2013). Much of NIST's work in the manufacturing sector lies within the NIST Engineering Laboratory (EL).

One of EL's manufacturing projects is *Prognostics and Health Management for Smart Manufacturing Systems* (PHM4SMS), which was initiated in 2013 (National Institute of Standards and Technology, 2014). The goal of this five-year effort is to develop and document methods, protocols, best practices, and tools to enable robust, real-time diagnostics and prognostics in manufacturing environments. These outputs will provide manufacturers

with uniform guidelines to identify the complex system, sub-system, and component interactions within smart manufacturing so they can understand the specific influences of each on process performance metrics and data integrity. Increased operational efficiency will be achieved through this greater understanding of the system, its constituent elements, and the multitude of relationships present.

1.2. PHM Needs and Challenges

Figure 1 shows a flowchart of the general process of PHM system development with certain standards listed for reference purposes. PHM system development begins with cost and dependability analyses to determine the components to monitor. The data management system is then initialized for collection, processing, visualization, and archiving of the maintenance data. Once the measurement techniques are established, the diagnostic and prognostic approaches are developed and tested to ensure that the desired goals are achieved. Finally, personnel are trained during the iterative process of system validation and verification before final system deployment.

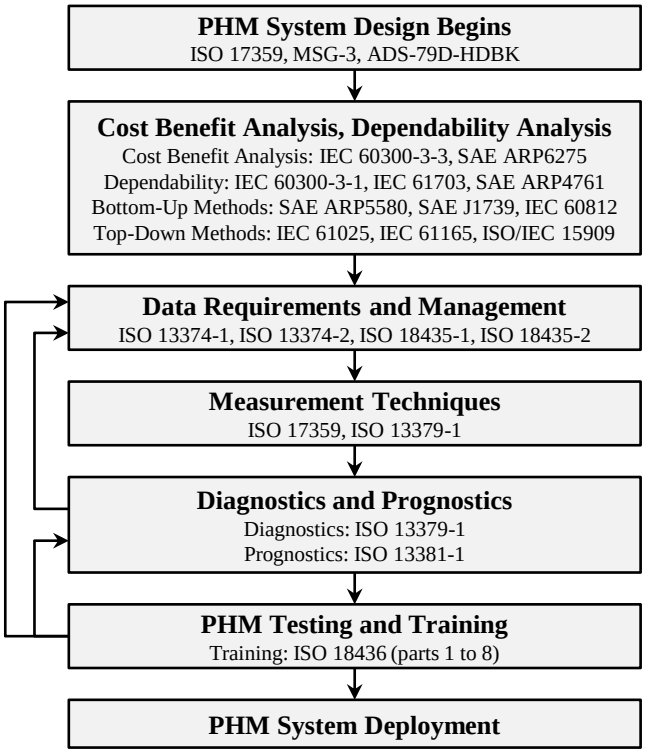


Figure 1. General PHM system development process and associated standards.

Several needs and challenges exist for PHM system development. PHM is dependent on maintenance-related data collection and processing for components or subsystems, so standards about data acquisition and processing are needed to influence the requirements for PHM systems development (United States Army, 2013). Standards for PHM are needed for harmonized terminology, consistency of the PHM methods and tools, and compatibility and interoperability of PHM technology. Standards also help provide guidance in the practical use and development of PHM techniques (Mathew, 2012). The creation of PHM systems is still difficult due to the inter-related tasks of design engineering, systems engineering, logistics, and user training (United States Army, 2013).

1.3. NIST PHM Efforts

PHM systems need to be developed, verified, and validated before implementation to enable improved decision-making for performance, safety, reliability, and maintainability of products and processes. However, standards appear to be lacking for PHM system development, data collection and analysis techniques, data management, system training, and software interoperability. The PHM4SMS project at NIST intends to help to serve a role in the development of such standards. The first step is to identify the existing pertinent standards, and this paper summarizes the results of such a review (Vogl, Weiss & Donmez, 2014).

2. PUBLISHED STANDARDS

Multiple organizations publish standards related to PHM for manufacturing products or processes. Table 1 lists the organizations that have published standards, while Table 2 (see Section 3) and Table 3 (see Appendix) categorize the developing or existing standards, respectively, related to PHM for manufacturing. All tables are organized according to topics based on the PHM process steps seen in Figure 1: 'Overview', 'Dependability analysis', 'Measurement techniques', 'Diagnostics and Prognostics', 'Data management', 'Training', and 'Applications'. If a standard has an 'X' mark in a corresponding general topic column within a table, then that standard is largely applicable within that category. Some of the standards outline broad approaches for PHM (marked in the 'Overview' category) or are specific in guidance for PHM within a given application (marked in the 'Applications' category). Other standards focus on dependability analysis, measurement techniques, diagnostics and/or prognostics, PHM data management, or training related to maintenance of systems. The lists of standards are not exhaustive, yet are comprehensive enough for those in the manufacturing fields.

As seen in Table 1, the standards were typically developed by a technical committee (TC) or subcommittee (SC) of various national and international organizations: the Air Transport Association (ATA), the International Electrotechnical Commission (IEC), the Institute of Electrical and Electronics Engineers (IEEE), the

International Organization for Standardization (ISO), the Machinery Information Management Open Standards Alliance (MIMOSA), SAE International, and the United States Army (US Army).

Table 1. PHM-related standards organizations.

Organization	Committee/ Subcommittee	Overview	Cost and Dependability analyses	Measurement techniques	Diagnostics and Prognostics	Data management	Training	Applications
ATA	MSG	X						X
IEC	56	X	X					
IEEE	RS	X						
ISO	TC 108/SC 2			X				
ISO	TC 108/SC 5	X		X	X	X	X	
ISO	TC 184/SC 4					X		
ISO	TC 184/SC 5				X	X		
ISO/IEC	JTC 1/SC 7		X					
MIMOSA	——					X		
SAE International	AQPIC		X					
SAE International	E-32	X			X			X
SAE International	G-11r	X	X					
SAE International	HM-1	X	X	X	X			X
US Army	Aviation Engineering	X		X	X			X

The following sections summarize the published standards in categories that are broad in scope: Overview, Dependability Analysis, Measurement Techniques, Diagnostics and Prognostics, and Data Management. Because they are outside the scope of NIST's current focus, Cost-, Training-, and Application-focused standards are not summarized.

2.1. Overview

Standards with general guidance about the creation of PHM systems are indicated under the 'Overview' category within Table 3. Such standards are a natural starting point during the creation of PHM systems, because these documents outline the factors influencing condition monitoring and provide guidance for the monitoring of components and/or sub-systems.

2.1.1. Manufacturing Industry

As the parent document of a group of standards that cover condition monitoring and diagnostics, ISO 17359 (International Organization for Standardization, 2011) was developed by ISO/TC 108/SC 5 ("Condition monitoring and diagnostics of machines") to provide the general procedures for setting up a condition monitoring program for all machines, e.g., the generic approaches to setting alarm criteria and carrying out diagnosis and prognosis. ISO 17359 outlines the condition monitoring procedure for a general manufacturing process, factors influencing condition monitoring, a list of issues affecting equipment criticality (e.g., cost of machine down-time, replacement cost), and a table of condition monitoring parameters (such as temperature, pressure, and vibration) for various machine types. ISO 17359 also presents multiple examples of tables showing the correlation of possible faults (e.g., air inlet blockage, seal leakage, and unbalance) with symptoms or parameter changes. Furthermore, ISO 17359 shows an example of a typical form for recording monitoring information.

2.1.2. Aircraft Industry

Another standard that provides guidance for PHM systems development is MSG-3, a document titled "Operator/Manufacturer Scheduled Maintenance Development." The Maintenance Steering Group (MSG) of the Air Transport Association (ATA) developed MSG-3, which is used for developing maintenance plans for aircraft, engines, and systems (Air Transport Association of America, 2013) before the aircraft enters service. MSG-3 is a top-down approach to determine the consequences (safety, operational, and economic) of failure, starting at the system level and working down to the component level (Adams, 2009). Failure effects are divided into five categories, and if the consequences of failure cannot be mitigated, then redesign becomes necessary. For example, the MSG-3 process led to mandatory design changes for the Boeing 787-8's in-flight control and lightning protection systems. Furthermore, the MSG-3 methodology helps improve safety while reducing maintenance-related costs up to 30 percent (Adams, 2009).

2.1.3. Military

Similar in scope to the standards just described, an Aeronautical Design Standard (ADS) Handbook (HDBK), ADS-79D-HDBK, was developed by the U.S. Army to describe the Army's condition-based maintenance (CBM) system for military aircraft systems (United States Army, 2013). CBM is the preferred maintenance approach for Army aircraft systems, yet ADS-79D-HDBK is broad enough for application in other industries to be included in the 'Overview' category of Table 3. The document provides guidance and standards for use by all Department of Defense (DoD) agencies in the development of CBM data acquisition, signal processing software, and data management. Furthermore, ADS-79D-HDBK is in the spirit of the reliability centered maintenance (RCM) methods previously used by the DoD to avoid the consequences of material failure. Failure mode, effects, and criticality analysis (FMECA) identifies where CBM should be utilized, but RCM is used to determine the most appropriate failure management strategy. Additionally, ADS-79D-

HDBK is supported by the Machinery Information Management Open Standards Alliance (MIMOSA), a United States association of industry and Government, and follows the information flow structure detailed in the ISO 13374 series (International Organization for Standardization, 2003, United States Army, 2013).

ADS-79D-HDBK defines CBM-related terms ('airworthiness', 'critical safety item', 'exceedance', etc.) and assists in the development of CBM systems for both legacy and new aircraft. Also, the standard describes the elements of a CBM system architecture with technical considerations for Army aviation in thirteen separate appendices (e.g., fatigue life management, flight test validation, vibration based diagnostics, and data integrity). These appendices help developers identify components to maintain, plan for data acquisition, perform fault testing, design the software and hardware elements, and validate CBM algorithms.

2.2. Dependability Analysis

One aspect of the generation of PHM systems outlined in Figure 1 is the determination of what components or subsystems should be redesigned, changed, or monitored due to their fault and/or failure potential. Typically, a dependability analysis involves the identification of the reliability, availability, and maintainability of the entire system, its subsystems, and its components (International Electrotechnical Commission, 2003).

Numerous methods exist to identify the failure modes of the system. Bottom-up (elements) methods are used to identify the failure modes at the component level, which are then used to determine the corresponding effect on higher-level system performance. On the other hand, top-down (functional) methods are used to identify undesirable system operations by starting from the highest level of interest (the top event) and proceeding to successively lower levels (International Electrotechnical Commission, 2003). Bottom-up dependability analysis methods include event tree analysis, failure mode and effects analysis (FMEA), and hazard and operability study (HAZOP), while top-down methods include fault tree analysis (FTA), Markov analysis, Petri net analysis, and reliability block diagrams (RBD).

2.2.1. General Guidance

IEC 60300-3-1 gives a general overview of the common dependability analysis techniques, including fault tree analysis, Markov analysis, Petri net analysis, and stress-strength analysis. IEC 60300-3-1 presents tables outlining the general applicability and characteristics of each method as well as concise summaries of each method (including benefits, limitations, and examples) in a separate informative annex (International Electrotechnical Commission, 2003). The methods can be categorized according to their purpose of either fault avoidance (e.g., stress-strength analysis), architectural analysis and dependability allocation (bottom-up methods, such as FMEA, or top-down methods, such as FTA), or estimation of measures for basic events (such as failure rate prediction). Analysis based on either a hardware (bottom-up), functional (top-down), or combination approach should be used to assess high risk items and provide corrective actions (United States Department of Defense, 1980).

Another standard that covers various dependability analyses is SAE ARP4761, an Aerospace Recommended Practice (ARP) that provides guidelines and methods of performing safety assessments for certification of civil aircraft (SAE International, 1996). Methods covered in SAE ARP4761 for safety assessment include FTA, dependence diagram (DD), Markov analysis, FMEA, and common cause analysis.

To support the quantification of dependability, the IEC technical committee 56 (Dependability) developed IEC 61703 to provide the mathematical expressions for reliability, availability, maintainability, and other maintenance terms (International Electrotechnical Commission, 2001). The expressions are grouped into classes for various items: non-repaired items, repaired items with zero time to restoration, and repaired items with non-zero time to restoration. Numerous equations are provided in IEC 61703 for the generic case of an exponentially distributed time to failure.

2.2.2. Bottom-Up Methods

FMEA

FMEA is a formal and systematic approach to identify potential failure modes of a system along with their causes and immediate and final effects on system performance (International Electrotechnical Commission, 2006a) through the usage of information about failure ("What has failed?") and its effects ("What are the consequences?") (SAE International, 2001). It is advantageous to perform FMEA early in the development of a product or process so that failure modes can be eliminated or mitigated as cost effectively as possible. FMEA can be used to identify failures (e.g., hardware, software, human performance) and improve reliability and maintainability via information for the development of diagnostic and maintenance procedures. FMEA has been modified for various purposes; failure modes, effects and criticality analysis (FMECA) is an extension of FMEA that uses a metric called *criticality* to rank the severity of failure modes (International Electrotechnical Commission, 2006a) as well as the probability of each failure mode (SAE International, 2001).

For example, SAE ARP5580 describes the procedure for how to perform FMEA. This procedure includes a basic methodology for the three FMEA classifications related to how the failure modes are postulated: functional FMEA (at

the conceptual design level), interface FMEA (before the detailed design of the interconnected subsystems), and detailed FMEA (performed when detailed designs are available) (SAE International, 2001). SAE ARP5580 can be used to assess the reliability of systems with increasing impact when FMEA is performed at increasing levels of detail during development of hardware or software. SAE ARP5580 provides many definitions of key terms (e.g., 'allocation', 'criticality', and 'fault tree') and other items typically included within FMEA. SAE ARP5580 provides ground rules (with an example), numbering conventions for functional FMEA to describe systems according to a hierarchy (subsystems, components, software, etc.) with well-defined inputs and outputs, and examples of severity classifications for military, aerospace, and automobile industries.

DFMEA and PFMEA

Another standard concerning FMEA is SAE J1739, which supports the development of an effective design FMEA (DFMEA) and a FMEA for manufacturing and assembly processes (PFMEA) (SAE International, 2009). Based on references (e.g., SAE ARP5580 and IEC 60812) and input from original equipment manufacturers (OEMs) and their suppliers, SAE J1739 includes current terms, requirements, ranking charts, and worksheets for the identification and mitigation of failure mode risks. Examples are given for a block or boundary diagram (for DFMEA), a process flow diagram (for PFMEA), and design and process FMEA worksheets related to the auto industry. Also, suggestions are given in tabulated form for design and process FMEA severity (S) evaluation criteria as well as those for occurrence (O) and detection (D) evaluation criteria. Even though the risk priority number (RPN) is defined as the product $S \times O \times D$, SAE J1739 warns that this number, which ranges from 1 to 1000, should not be used as the sole metric for risk evaluation via thresholding.

FMEA and FMECA

Another standard that gives guidance to produce successful FMEA and FMECA is IEC 60812, which was developed by the IEC technical committee 56 (Dependability) (International Electrotechnical Commission, 2006a). IEC 60812 is a standard that provides steps, terms, criticality measures (potential risk, risk priority number, criticality matrix), failure modes, basic principles, procedures, and examples for FMEA and FMECA. IEC 60812 advises that while FMECA may be a very cost-effective method for assessing failure risks, a probability risk analysis (PRA) is preferable to a FMECA; FMECA should not be the only basis for judging risks, especially since RPNs have deficiencies such as inadequate scaling, as discussed in SAE J1739. Also, FMEA has limitations in that it is difficult and tedious to apply to complex systems with multiple functions (International Electrotechnical Commission, 2006a).

2.2.3. Top-Down Methods

Fault Tree Analysis (FTA)

FTA is a technique that is helpful in overcoming the current limitations of FMEA (SAE International, 2001). FTA is a deductive method used to determine the causes that can lead to the occurrence of a defined outcome, called the 'top event' (International Electrotechnical Commission, 2006b). FTA achieves this goal through use of a fault tree. Construction of the tree is a top-down process that continually approaches the desired lower level of mechanism and mode. The lowest possible level contains the primary (bottom) events, the individual causes of potential failures or faults (International Electrotechnical Commission, 2006b). Thus, FTA identifies potential problems caused by design, operational stresses, and flaws in product manufacturing processes. Hence, fault trees should be developed early during system design and continue throughout the development of a product (International Electrotechnical Commission, 2006b).

To enable the use of fault tree analysis, the IEC technical committee 56 developed IEC 61025, which addresses the two approaches to FTA: a qualitative or logical approach (Method A), used largely in the nuclear industry, and a quantitative or numerical approach (Method B) that results in a quantitative probability of the occurrence of a top event within manufacturing and other industries (International Electrotechnical Commission, 2006b). IEC 61025 describes FTA with its definitions (e.g., 'top event', 'gate', and 'event'), steps (fault tree construction, analysis, reporting, etc.), and fault tree symbols (for static and dynamics gates). IEC 61025 provides the mathematics for reliability of series and parallel (redundant) systems, which uses probabilistic data at the component level from reliability or actual field test data to determine the probability of the occurrence of the 'top event'.

Markov Analysis

Markov analysis is another method to determine the dependability and safety of systems. The IEC technical committee 56 produced IEC 61165, a standard that gives an overview of the Markov technique (International Electrotechnical Commission, 2006c). Markov techniques use state transition diagrams to represent the temporal behavior of a system, which is a connected number of elements, each of which has only one of two states: up or down. The entire system transitions from one state to another as the system elements fail or are restored according to defined rates. IEC 61165 uses symbols from IEC 60050 ('International Electrotechnical Vocabulary') but defines other fundamental terminology (e.g., 'up state' and 'down state'), symbols (circles, rectangles, etc.), and mathematical techniques (e.g., via ordinary differential equations and Laplace transforms). The standard contains examples for the

homogeneous Markov technique, in which the state transition rates are assumed to be time-independent (International Electrotechnical Commission, 2006c). IEC 61165 shows that the differences between the expressions for reliability, maintainability, and availability arise from the different state transition diagrams used to create the equations. Maintenance strategies can be modeled with Markov techniques, while other techniques such as fault tree analysis (FTA) and reliability block diagrams (RBDs) do not account for complex maintenance strategies.

Petri Net Analysis

Since their creation in 1962, Petri nets have been used to describe, design, and maintain a wide range of systems and processes in industries including aerospace, banking, manufacturing systems, and nuclear power systems (International Organization for Standardization & International Electrotechnical Commission, 2004). Petri nets are a rigorous method to mathematically describe processes based on basic set theory (Truss, 1998). Furthermore, Petri nets can be used to generate Markov models. In the 1980s, Petri nets were extended to Higher-level Petri nets (HLPNs) to model discrete-event systems. HLPNs were also used to advance the use of Petri nets for complex systems, analogous to the use of high-level programming languages to overcome challenges with assembly languages.

To aid the use of HLPNs and facilitate the development of Petri net software tools, the ISO/IEC 15909-1 standard was developed by SC 7 ('Software and system engineering') of JTC 1 ('Information technology'), a Joint Technical Committee (JTC) composed of ISO and IEC members (International Organization for Standardization & International Electrotechnical Commission, 2004). ISO/IEC 15909-1 defines a mathematical semantic model, an abstract mathematical syntax for annotations, and a graphical notation for High-level Petri nets (International Organization for Standardization & International Electrotechnical Commission, 2004). ISO/IEC 15909-1 defines terms (such as 'arc', 'multiset', 'Petri net', 'token', 'transition', etc.) and mathematical conventions needed for High-level Petri nets and provides the formal concepts of marking, enabling, and transition rules needed for HLPN graphs (HLPNGs) that represent complex processes within manufacturing and other industries. ISO/IEC 15909-2 defines the transfer format, the Petri Net Markup Language (PNML), to support the exchange of HLPNs (International Organization for Standardization & International Electrotechnical Commission, 2011).

2.3. Measurement Techniques

Dependability analysis, whether top-down or bottom-up or some combination thereof, is used to identify the failure modes of the system and help manufacturers to determine which risks should be mitigated or eliminated. If a failure mode must exist, being unavoidable for system operation, then the failure mode may be monitored or predicted via diagnostics and prognostics with sensors and established measurement and analysis techniques. The system designer must be aware of the various measurement techniques and their preferred uses based on the accepted experience of others.

Several standards contain explicit guidelines on the use of measurement techniques for PHM. This section summarizes those particular standards indicated under the 'Measurement techniques' category within Table 3. However, due to the detailed nature and variety of measurement techniques, this section covers only the standards that are relatively general in scope and application for manufacturing.

For example, Annex B of ISO 17359 contains nine tables of guidance for measurement techniques for various systems, including generators, fans, engines, and pumps (International Organization for Standardization, 2011). The tables relate the possible faults for each system to the associated measureable symptoms. For example, ISO 17359 reveals that the bearing unbalance of an electric motor affects the vibration directly, but only impacts the other detectable symptoms tangentially. Such tables are essential for understanding the basic physical consequences of system faults to aid in the selection and positioning of sensors. Similarly, Annex D of ISO 13379-1 relates measurement techniques and numerous diagnostic models in tabular form (International Organization for Standardization, 2012b). The combination of the information from ISO 17359 and ISO 13379-1 helps both novices and experts in PHM to determine the measurement types and associated diagnostic techniques for a given system fault. For example, a bearing unbalance could be detected via vibration monitoring (according to ISO 17359) and analyzed via a subsequent data-driven statistical method (according to ISO 13379-1).

2.4. Diagnostics and Prognostics

Diagnostics is the determination of the current condition of a component or system, and prognostics is the predictive ability of future performance degradation and expected failures (SAE International, 2008). The following subsections summarize those particular standards indicated under the 'Diagnostics and Prognostics' category within Table 3. The number of standards dedicated to diagnostics and prognostics is fairly small, offering a significant opportunity for standards development.

2.4.1. Diagnostics

One recently-published standard aids the diagnostics of general PHM processes; ISO 13379-1 was created to aid the condition monitoring of industrial machines including turbines, compressors, pumps, generators, electrical motors, blowers, gearboxes, and fans (International Organization for Standardization, 2012b). ISO 13379-1, which was prepared under SC 5 (Condition monitoring and diagnostics of

machines) of ISO/TC 108 (Mechanical vibration, shock and condition monitoring), outlines the nine generic steps for diagnostics, composed of the union of FMEA or FMECA, as outlined in IEC 60812, and failure mode symptoms analysis (FMSA) methodology outlined in ISO 13379-1. FMSA is essentially a modification of a FMECA process that focuses on the selection of the most appropriate detection and monitoring techniques and strategies. The process results in a monitoring priority number (MPN) for each failure mode. The MPN is the product of four numbers representing the confidence (each rated from 1 to 5) of detection, severity, diagnosis, and prognosis for the given failure mode. The highest MPN value indicates the most suitable technique for detection, diagnostics, and prognostics of the associated failure mode (International Organization for Standardization, 2012b).

ISO 13379-1 also compares the strengths and weaknesses of data-driven diagnostic approaches (e.g., neural network, logistic regression, and support vector machine) and knowledge-based diagnostic approaches (e.g., causal tree and first principles). The last step in the diagnostic process is a formal diagnostic report, such as the example given in Annex E of ISO 13379-1, which includes information about the event, its diagnosis, symptoms, failure modes, and recommendations for corrective action and fault avoidance.

2.4.2. Prognostics

Other standards provide guidance for prognostics, because there is currently no precise procedure or standard methodology. Fault prognostics require prior knowledge of the probable failure modes, the anticipated future activities of the machine, and the relationships between failure modes and operating conditions (International Organization for Standardization, 2004).

To facilitate the development of prognostics within general PHM processes, ISO 13381-1 outlines general guidelines, approaches, and concepts for prognostics (International Organization for Standardization, 2004). Terms such as prognosis (an estimation of time to failure and associated risk), confidence level, root cause, and estimated time to failure (ETTF) are defined in ISO 13381-1. The standard also outlines the four basic phases of prognosis: pre-processing, existing failure mode prognosis, future failure mode prognosis, and post-action prognosis. ISO 13381-1 states that the trip set point used for thresholding to prevent damage or failure is a parameter value, normally determined from standards, manufacturers' guidelines, and experience. Other thresholds, such as alert and alarm limits, are set at values below the trip set point to initiate maintenance. Once a fault has been detected based on a failure mode behavior model (FMECA, FTA, etc.), the estimated time to failure (ETTF) needs to be determined by expert opinion and/or empirical methods (International Organization for Standardization, 2004).

2.5. Data Management

Monitoring the condition of machines is not an easy task because the integration of various PHM software is typically not 'plug-and-play' (International Organization for Standardization, 2003). This section summarizes several standards that guide the management of PHM data and, hence, the integration of various PHM software via the transfer of standardized data formats.

ISO 13374-1 provides the basic requirements for open software specifications to facilitate the transfer of data among various condition monitoring software, regardless of platform or hardware protocols (International Organization for Standardization, 2003). ISO 13374-1 establishes the general guidelines, including the requirement of an 'open machine condition monitoring information schema architecture as an underlying framework' (International Organization for Standardization, 2003). Vendor-independent extensible markup language (XML) schema and protocols can be used for the network exchange of PHM information. In accordance with ISO 13374, the Machinery Information Management Open Systems Alliance (MIMOSA) published a conceptual schema called the Common Relational Information Schema (CRIS) in XML schema and other formats. The CRIS has been used in the condition monitoring industry to integrate information from many systems (MIMOSA, 2006).

ISO 13374-2 provides details of the methodology and requirements for data processing within condition monitoring and diagnostics (CM&D) systems. ISO 13374-2 describes all the data objects, types, relationships, etc. required for a CM&D information architecture (International Organization for Standardization, 2007). ISO 13374-2 provides an informative annex about the unified modeling language (UML), XML, and Middleware services. Finally, MIMOSA publishes an open CM&D information specification known as the MIMOSA Open Systems Architecture for Enterprise Application Integration (OSA-EAI™), which is compliant with the requirements outlined in ISO 13374-1 and ISO 13374-2 and free for download (MIMOSA, 2013). MIMOSA also publishes an open CM&D specification known as the MIMOSA Open Systems Architecture for Condition Based Maintenance (OSA-CBM™), which is based on OSA-EAI™, enabling integration of systems from various suppliers (International Organization for Standardization, 2007).

ISO 18435-1 gives an overview of the elements and rules of an integration modeling method to describe a manufacturing application's requirements for integration of an automation application with other applications, e.g., diagnostics, prognostics, capability assessment, and maintenance applications with production and control applications (International Organization for Standardization, 2009). The method is based upon the Application Domain Integration Diagram (ADID), which facilitates the transfer

of information among domains of the manufacturing process. The domains include the processing blocks of ISO 13374, such as the Data Monitoring block or the State Detection block. ISO 18435-1 defines terms (e.g., 'integration' and 'interaction') and provides examples of exchanged information among domains.

ISO 18435-2 defines the application interaction matrix element (AIME) and application domain matrix element (ADME) structures and relationships, including the steps to construct an ADME for support by a set of AIMEs (International Organization for Standardization, 2012a). An AIME represents a set of capabilities provided by a set of manufacturing resources of an application. An ADME is a means to model the information exchanges between applications, being constructed from interoperability profiles referenced in AIMEs. ISO 18435-2 outlines the XML schema for the headers and bodies that comprise AIMEs and ADMEs. AIME bodies consist of context and conveyance sections, and ADME bodies consist of context, conveyance, and content sections. ISO 18435-2 also contains formal definitions of the ADME/AIME schemas in informative annexes (International Organization for Standardization, 2012a).

3. CURRENT STANDARDS DEVELOPMENT

New standards and revisions to existing standards related to PHM are currently under development, as seen in Table 2. This section summarizes the scopes of these standards.

Table 2. PHM-related standards under development.

Organization	Committee/ Subcommittee	Standard	1st Edition / Revision?	Overview	Dependability analysis	Diagnostics and Prognostics	Data management
SAE Int.	G-11r	ARP6204	1st Edition	X			
SAE Int.	HM-1	ARP6268	1st Edition	X			
SAE Int.	HM-1	ARP6407	1st Edition	X			
SAE Int.	HM-1	ARP6883	1st Edition	X			
IEEE	RS	P1856	1st Edition	X			
ISO/IEC	JTC 1/SC 7	ISO/IEC 15909-2	1st Edition		X		
ISO	TC 108/SC 5	ISO 13379-2	1st Edition			X	
ISO	TC 108/SC 5	ISO 13381-1	Revision			X	
ISO	TC 108/SC 5	ISO 18129	1st Edition			X	
ISO	TC 184/SC 5	ISO 22400-1	1st Edition			X	
ISO	TC 184/SC 5	ISO 22400-2	1st Edition			X	
ISO	TC 184/SC 5	ISO 18435-3	1st Edition				X
SAE Int.	HM-1	ARP6290	1st Edition				X

3.1. Overview

Currently, SAE International is developing SAE ARP6204, a standard for "Condition Based Maintenance (CBM) Recommended Practices," under the G-11r Reliability Committee. The scope of the document is to outline a path for an organization to implement a CBM approach to maintenance, including practices regarding both CBM design and field equipment support (SAE International, 2013). The G-11r Reliability Committee has benchmarked the CBM framework and performance specifications and is developing a formal application specification (Zhou, Bo & Wei, 2013).

Other SAE International standards are under development in the HM-1 Integrated Vehicle Health Management (IVHM) Committee. Guidance is lacking for the systems engineering aspects of IVHM design; SAE ARP6407 will help to fill this gap by providing technology-independent guidance for the design of IVHM systems (SAE International, 2014a). Furthermore, SAE ARP6883 will provide guidelines for writing IVHM requirements for aerospace systems, and SAE ARP6268 will help improve coordination and communication between manufacturers and suppliers.

Another broad standard under development is IEEE P1856 - "Standard Framework for Prognostics and Health Management of Electronic Systems" (IEEE Standards Association, 2013). In 2012, the IEEE Standards Board approved the new standard development project to produce IEEE P1856, which is sponsored by the Reliability Society (IEEE-RS) (IEEE Reliability Society, 2014). The working group meets regularly to prepare a draft for ballot in 2014 (IEEE Reliability Society, 2014). Even though this standard is being developed by IEEE, the intent is for it to have broad applicability in mechanical structures, civil structures, nuclear technology, and aeronautics (The Center for Advanced Life Cycle Engineering (CALCE), 2013).

3.2. Dependability Analysis

The first edition of ISO/IEC 15909-3 is under development by ISO/IEC JTC 1/SC 7 to aid the use of High-level Petri nets (International Organization for Standardization & International Electrotechnical Commission, 2014). ISO/IEC 15909-3, expected to be the last part of the ISO/IEC 15909 series, will address the techniques for modularity and extensions of High-level Petri nets for dependability analysis of PHM systems.

3.3. Diagnostics and Prognostics

ISO 13379-2 ('Data-driven applications') will aid the condition monitoring of industrial machines via diagnostics and is currently in the committee draft stage within ISO/TC 108/SC 5. Also, ISO 13381-1 is now at the committee draft stage while being updated to advance prognostics within PHM systems. Furthermore, within the same subcommittee, a new standard, ISO 18129, is in the draft international stage to address 'approaches for performance diagnosis' (International Organization for Standardization, 2014).

The ISO 22400 series of standards are also being developed by ISO/TC 184/SC 5 to guide the creation, computation, measurement, utilization, and maturation of key performance indicators (KPIs) within the manufacturing operations management (MOM) domain (International Organization for Standardization, 2013). KPIs are the most useful measures for monitoring and evaluating the performance of a production-oriented enterprise to help industries meet their performance targets in an intelligent manner (International Organization for Standardization, 2013). Because KPIs are serviced by effective PHM systems, standards related to KPIs could easily influence the diagnostic and prognostic aspects of PHM systems. NIST personnel are active in the development of the ISO 22400 standard series.

3.4. Data Management

SAE ARP6290, under development in the HM-1 Committee, will provide guidance for the creation of optimum architectures for IVHM that are in line with the organization's business goals and objectives. SAE ARP6290 will incorporate suggestions from ISO 13374 into specific guidelines for IVHM architecture development (SAE International, 2014b).

Future improvements to ATA MSG-3 (Air Transport Association of America, 2013), used for developing maintenance plans for aircraft, engines, and systems, will involve an existing data format specification known as ATA SPEC2000, a comprehensive set of e-Business specifications, products, and services that help to overcome the supply chain challenges in the aircraft industry (Air Transport Association of America, 2012). ATA SPEC2000 helps aircraft manufacturers with information exchange in order to have statistically significant data for optimizing and developing maintenance programs.

4. CONCLUSIONS

The National Institute of Standards and Technology conducted a survey of PHM-related standards to determine the industries and needs addressed by such standards, the extent of these standards, and any similarities as well as potential gaps among the documents. This effort revealed that standards exist that are related to all aspects of the development of prognostics and health management systems: general overview, dependability analysis, measurement techniques, diagnostic analysis, prognostic analysis, data management, performance metrics, and personnel training. Some standards were focused on providing guidance for specific applications, yet still broad enough for general application across industries. Other standards were more focused on a specific product or process within a target industry.

Based on the lessons learned from the PHM-related standards, recommendations can be made for the development of future PHM standards:

- The 'overview' standards cover numerous domains yet could be updated and harmonized by the respective organizations to provide better consolidation among the separate standards, providing for a more generally approved PHM process across disciplines.
- The 'dependability analysis' standards could be extended by combining the KPI standards under development with a dependability method to provide a bridge of guidance between design and business decisions for manufacturing systems and systems of systems.
- The 'diagnostics and prognostics' standards are lacking, due in part to the difficult nature of reliable diagnostics and prognostics techniques across various industries. However, the existing standards are still valuable for industry. Collaborations among PHM experts are recommended for the generation of new standards for diagnostics and prognostics that fill high-priority gaps for manufacturing systems. Priorities will be established at an upcoming industry workshop held at NIST in November 2014.
- The 'data management' standards appear to be thorough and consistent among each other, providing generic structures for PHM data and control flow. Extension to a 'digital factory' could be reported in future editions of these standards.

Consequently, NIST is exploring the development of methods and supporting standards for PHM of manufacturing systems and systems of systems.

ACKNOWLEDGEMENT

The authors thank Patrice Boulanger (Standards Coordination Office, NIST) for her pivotal help in the attainment of the standards used for this work.

REFERENCES

C. Adams. (2009) Understanding MSG-3. *Aviation Today*. Available: http://www.aviationtoday.com/am/repairstations/Understanding-MSG-3_33062.html

Air Transport Association of America (2012). *SPEC2000*: http://www.spec2000.com/

Air Transport Association of America (2013). MSG-3: Operator/Manufacturer Scheduled Maintenance Development, Volume 1 – Fixed Wing Aircraft

R. Bostelman, J. Teizer, S. J. Ray, M. Agronin, & D. Albanese (2014). Methods for improving visibility measurement standards of powered industrial vehicles. *Safety science*, vol. 62, pp. 257-270.

K. A. Hunten, A. Barnard Feeney, & V. Srinivasan (2013). Recent advances in sharing standardized STEP composite structure design and manufacturing information. *Computer-Aided Design*, vol. 45, pp. 1215-1221.

IEEE Reliability Society (2014). *IEEE Standards Development*: http://rs.ieee.org/ieee-standards-development.html

IEEE Standards Association (2013). *P1856 - Standard Framework for Prognostics and Health Management of Electronic Systems*: http://standards.ieee.org/develop/project/1856.html

International Electrotechnical Commission (2001). IEC 61703 - Mathematical expressions for reliability, availability, maintainability and maintenance support terms

International Electrotechnical Commission (2003). IEC 60300-3-1 - Dependability management – Part 3-1: Application guide – Analysis techniques for dependability – Guide on methodology

International Electrotechnical Commission (2006a). IEC 60812 - Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)

International Electrotechnical Commission (2006b). IEC 61025 - Fault tree analysis (FTA)

International Electrotechnical Commission (2006c). IEC 61165 - Application of Markov techniques

International Organization for Standardization (2003). ISO 13374-1:2003 - Condition monitoring and diagnostics of machines − Data processing, communication and presentation − Part 1: General guidelines

International Organization for Standardization (2004). ISO 13381-1:2004 - Condition monitoring and diagnostics of machines − Prognostics − Part 1: General guidelines

International Organization for Standardization & International Electrotechnical Commission (2004). ISO/IEC 15909-1:2004 - Software and system engineering − High-level Petri nets − Part 1: Concepts, definitions and graphical notation

International Organization for Standardization (2007). ISO 13374-2:2007 - Condition monitoring and diagnostics of machines − Data processing, communication and presentation − Part 2: Data processing

International Organization for Standardization (2009). ISO 18435-1:2009 - Industrial automation systems and integration − Diagnostics, capability assessment and maintenance applications integration − Part 1: Overview and general requirements

International Organization for Standardization (2011). ISO 17359:2011 - Condition monitoring and diagnostics of machines − General guidelines

International Organization for Standardization & International Electrotechnical Commission (2011). ISO/IEC 15909-2:2011 - Software and system engineering − High-level Petri nets − Part 2: Transfer format

International Organization for Standardization (2012a). ISO 18435-2:2012 - Industrial automation systems and integration − Diagnostics, capability assessment and maintenance applications integration − Part 2: Descriptions and definitions of application domain matrix elements

International Organization for Standardization (2012b). ISO 13379-1:2012 - Condition monitoring and diagnostics of machines − Data interpretation and diagnostics techniques − Part 1: General guidelines

International Organization for Standardization (2013). ISO/WD 22400-1 - Manufacturing operations management − Key performance indicators − Part 1: Overview, concepts and terminology

International Organization for Standardization (2014). ISO/DIS 18129 - Condition monitoring and diagnostics of machines − Approaches for performance diagnosis

International Organization for Standardization & International Electrotechnical Commission (2014). ISO/IEC CD 15909-3 - Software and system engineering − High-level Petri nets − Part 3: Petri Net Extensions

K. B. Lee, E. Y. Song, & P. S. Gu (2012). Integration of MTConnect and Standard-Based Sensor Networks for Manufacturing Equipment Monitoring. *ASME 2012 International Manufacturing Science and Engineering Conference*, pp. 841-848

H. Ludwig & E. Spiegel. (2014, January 20, 2014) America's Real Manufacturing Advantage. *Strategy+Business*. 11.

J. Marvel & R. Bostelman (2013). Towards mobile manipulator safety standards. *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, pp. 31-36

S. Mathew (2012). *PHM Standards - IEEE PHM Standard*: https://www.phmsociety.org/sites/phmsociety.org/files/PHM%20Society%20Standards%20Research%20Panel_IEEE%20PHM.pdf

MIMOSA (2006). *MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI) Technical Architecture Summary* http://www.mimosa.org/sites/default/files/TechDocs/OSA-EAI_Technical_Architecture_Summary_Dec_2006.pdf

MIMOSA (2013). *MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI)*: http://www.mimosa.org/?q=node/300

National Institute of Standards and Technology (2014). *Prognostics and Health Management for Smart Manufacturing Systems*: http://www.nist.gov/el/isd/ks/phm4sms.cfm

SAE International (1996). ARP4761 - Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment

SAE International (2001). ARP5580 - Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications

SAE International (2008). AIR5871 - Prognostics for Gas Turbine Engines

SAE International (2009). J1739 - Potential Failure Mode and Effects Analysis in Design (Design FMEA), Potential Failure Mode and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA)

SAE International (2013). *ARP6204 - Condition Based Maintenance (CBM) Recommended Practices*: http://standards.sae.org/wip/arp6204/

SAE International (2014a). *Integrated Vehicle Health Management Design Guidelines*: http://standards.sae.org/wip/arp6407/

SAE International (2014b). *ARP6290 - Guidelines for the Development of Architectures for Integrated Vehicle Health Management Systems*: http://standards.sae.org/wip/arp6290/

The Center for Advanced Life Cycle Engineering (CALCE) (2013). *CALCE PHM News - August 2013*: http://www.prognostics.umd.edu/WhatsNew/2013/calce-phm-aug2013.html

J. K. Truss (1998). *Discrete Mathematics for Computer Scientists*, 2nd ed.: Addison-Wesley

United States Army (2013). ADS-79D-HDBK - Aeronautical Design Standard Handbook for Condition Based Maintenance Systems for US Army Aircraft

United States Department of Defense (1980). MIL-STD-1629A - Procedures for performing a failure mode, effects and criticality analysis

G. W. Vogl, B. A. Weiss, & M. A. Donmez (2014). *Standards Related to Prognostics and Health Management (PHM) for Manufacturing*. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA, NISTIR 8012. doi: 10.6028/NIST.IR.8012

Y. Zhou, J. Bo, & T. Wei (2013). A review of current prognostics and health management system related standards. *Chemical Engineering Transactions*, vol. 33, pp. 277-282. doi: 10.3303/CET1333047

BIOGRAPHIES

Dr. Gregory W. Vogl is a Mechanical Engineer at the National Institute of Standards and Technology (NIST) located in Gaithersburg, Maryland. He received his B.S. (2000), M.S. (2003), and Ph.D. (2006) degrees in Engineering Mechanics from Virginia Tech, Virginia, USA. Currently, Greg is a member of the *Prognostics and Health Management for Smart Manufacturing Systems* (PHM4SMS) project, which seeks to develop a methodology, protocols, and reference datasets to enable robust real-time diagnostics and prognostics for smart manufacturing systems. Previously, he designed, fabricated, and experimented on microelectromechanical systems as a National Research Council Postdoctoral Researcher at NIST. He then joined the Production Systems Group, in which he worked on machine tool metrology and standards development. His interests include machine tool spindle health, diagnostic and prognostic methods, nonlinear dynamics, engineering mechanics, and metrology.

Dr. Brian A. Weiss has a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the *Smart Manufacturing Operations Planning and Control* program and the Project Leader of the *Prognostics and Health Management for Smart Manufacturing Systems* project within the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). Prior to his leadership roles in the SMOPAC program and the PHM4SMS project, he spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles; tactical applications operating on Android devices; advanced soldier sensor technologies; free-form, two-way, speech-to-speech translation devices for tactical use; urban search and rescue robots; and bomb disposal robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 & 2008), and the Jacob Rabinow Applied Research Award (2006).

Dr. Alkan Donmez is currently the Group Leader of the Production Systems Group as well as the Program Manager for the *Measurement Science for Additive Manufacturing* program in the NIST Engineering Laboratory. He has been with NIST for more than 25 years conducting and supervising research in advanced manufacturing sciences, including machine tool performance modeling and metrology, machining process metrology, as well as the recent efforts in metal-based additive manufacturing (AM). He has actively participated in national and international standard committees, developing machine tool performance testing standards, for more than 20 years. He has published more than 70 technical papers and reports in the area of machine tool metrology and manufacturing sciences. He has received various awards for his technical contributions, including R&D100, Applied Research Award of NIST, and Department of Commerce Silver and Bronze Medals.

APPENDIX

Table 3. Standards related to PHM for manufacturing.

Organization	Committee/ Subcommittee	Standard	Year Issued	Title	Overview	Cost and Dependability analyses	Measurement techniques	Diagnostics and Prognostics	Data management	Training	Applications
ATA	MSG	ATA MSG-3	2013	MSG-3: Operator/Manufacturer Scheduled Maintenance Development, Volume 1 – Fixed Wing Aircraft	X						X
IEC	56	IEC 61703	2001	Mathematical expressions for reliability, availability, maintainability and maintenance support terms	X						
ISO	TC 108/SC 5	ISO 13372	2012	Condition monitoring and diagnostics of machines – Vocabulary	X						
ISO	TC 108/SC 5	ISO 17359	2011	Condition monitoring and diagnostics of machines − General guidelines	X		X				
SAE	E-32	ARP1587B	2007	Aircraft Gas Turbine Engine Health Management System Guide	X						X
US Army	Aviation Engineering	ADS-79D-HDBK	2013	Aeronautical Design Standard Handbook for Condition Based Maintenance Systems for US Army Aircraft	X		X	X			X
IEC	56	IEC 60300-3-1	2003	Dependability management – Part 3-1: Application guide – Analysis techniques for dependability – Guide on methodology		X					
IEC	56	IEC 60300-3-3	2004	Dependability management – Part 3-3: Application guide – Life cycle costing		X					
IEC	56	IEC 60812	2006	Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)		X					
IEC	56	IEC 61025	2006	Fault tree analysis (FTA)		X					
IEC	56	IEC 61165	2006	Application of Markov techniques		X					
SAE	AQPIC	J1739	2009	Potential Failure Mode and Effects Analysis in Design (Design FMEA), Potential Failure Mode and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA)		X					
SAE	HM-1	ARP6275	2014	Determination of Cost Benefits from Implementing an Integrated Vehicle Health Management System		X					
SAE	G-11r	ARP5580	2001	Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications		X					
SAE	S-18	ARP4761	1996	Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment		X					X
ISO/IEC	JTC 1/SC 7	ISO/IEC 15909-1	2004	Software and system engineering − High-level Petri nets − Part 1: Concepts, definitions and graphical notation		X					
ISO/IEC	JTC 1/SC 7	ISO/IEC 15909-2	2011	Software and system engineering − High-level Petri nets − Part 2: Transfer format		X					
ISO	TC 108/SC 2	ISO 13373-1	2002	Condition monitoring and diagnostics of machines − Vibration condition monitoring − Part 1: General procedures			X				
ISO	TC 108/SC 2	ISO 13373-2	2005	Condition monitoring and diagnostics of machines − Vibration condition monitoring − Part 2: Processing, analysis and presentation of vibration data			X				
ISO	TC 108/SC 5	ISO 18434-1	2008	Condition monitoring and diagnostics of machines – Thermography – Part 1: General procedures			X				
ISO	TC 108/SC 5	ISO 20958	2013	Condition monitoring and diagnostics of machine systems – Electrical signature analysis of three-phase induction motors			X				
ISO	TC 108/SC 5	ISO 22096	2007	Condition monitoring and diagnostics of machines – Acoustic emission			X				
ISO	TC 108/SC 5	ISO 29821-1	2011	Condition monitoring and diagnostics of machines – Ultrasound – Part 1: General guidelines			X				
ISO	TC 108/SC 5	ISO 13379-1	2012	Condition monitoring and diagnostics of machines − Data interpretation and diagnostics techniques − Part 1: General guidelines				X			
ISO	TC 108/SC 5	ISO 13381-1	2004	Condition monitoring and diagnostics of machines − Prognostics − Part 1: General guidelines				X			

SAE	E-32	AIR5871	2008	Prognostics for Gas Turbine Engines				X			X
ISO	TC 184/SC 4	ISO 15531-1	2004	Industrial automation systems and integration – Industrial manufacturing management data – Part 1: General overview					X		
ISO	TC 184/SC 4	ISO 15531-42	2005	Industrial automation systems and integration – Industrial manufacturing management data – Part 42: Time Model					X		
ISO	TC 184/SC 4	ISO 15531-43	2006	Industrial automation systems and integration – Industrial manufacturing management data – Part 43: Manufacturing flow management data: Data model for flow monitoring and manufacturing data exchange					X		
ISO	TC 184/SC 4	ISO 15531-44	2010	Industrial automation systems and integration – Industrial manufacturing management data – Part 44: Information modelling for shop floor data acquisition					X		
ISO	TC 184/SC 4	ISO 15926-1	2004	Industrial automation systems and integration − Integration of life-cycle data for process plants including oil and gas production facilities − Part 1: Overview and fundamental principles					X		
ISO	TC 184/SC 4	ISO 15926-2	2003	Industrial automation systems and integration − Integration of life-cycle data for process plants including oil and gas production facilities − Part 2: Data model					X		
ISO	TC 108/SC 5	ISO 13374-1	2003	Condition monitoring and diagnostics of machines − Data processing, communication and presentation − Part 1: General guidelines					X		
ISO	TC 108/SC 5	ISO 13374-2	2007	Condition monitoring and diagnostics of machines − Data processing, communication and presentation − Part 2: Data processing					X		
ISO	TC 108/SC 5	ISO 13374-3	2012	Condition monitoring and diagnostics of machines − Data processing, communication and presentation − Part 3: Communication					X		
ISO	TC 184/SC 5	ISO 18435-1	2009	Industrial automation systems and integration – Diagnostics, capability assessment and maintenance applications integration – Part 1: Overview and general requirements					X		
ISO	TC 184/SC 5	ISO 18435-2	2012	Industrial automation systems and integration − Diagnostics, capability assessment and maintenance applications integration − Part 2: Descriptions and definitions of application domain matrix elements					X		
ISO	TC 108/SC 5	ISO 18436-1	2012	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 1: Requirements for assessment bodies and the assessment process						X	
ISO	TC 108/SC 5	ISO 18436-2	2003	Condition monitoring and diagnostics of machines – Requirements for training and certification of personnel – Part 2: Vibration condition monitoring and diagnostics						X	
ISO	TC 108/SC 5	ISO 18436-3	2012	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 3: Requirements for training bodies and the training process						X	
ISO	TC 108/SC 5	ISO 18436-4	2008	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 4: Field lubricant analysis						X	
ISO	TC 108/SC 5	ISO 18436-5	2012	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 5: Lubricant laboratory technician/analyst						X	
ISO	TC 108/SC 5	ISO 18436-6	2008	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 6: Acoustic emission						X	
ISO	TC 108/SC 5	ISO 18436-7	2008	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 7: Thermography						X	
ISO	TC 108/SC 5	ISO 18436-8	2013	Condition monitoring and diagnostics of machines – Requirements for qualification and assessment of personnel – Part 8: Ultrasound						X	
SAE	S-18	ARP4754A	2010	Guidelines for Development of Civil Aircraft and Systems							X

A Self-Aware Machine Platform in Manufacturing Shop Floor Utilizing MTConnect Data

Linxia Liao[1], Raj Minhas[2], Arvind Rangarajan[3], Tolga Kurtoglu[4], Johan de Kleer[5]

[1,2,4,5] *Palo Alto Research Center, Palo Alto, CA, 94304, USA*
linxia.liao@parc.com
raj.minhas@parc.com
tolga.kurtoglu@parc.com
dekleer@parc.com

[3] *General Electric Global Research, San Ramon, CA, 94583, USA*
arvind.rangarajan@ge.com

ABSTRACT

We propose a framework of self-aware machines based on data collected using the MTConnect protocol. Beyond existing applications of OEE (Overall Equipment Effectiveness) reporting, the proposed framework integrates multiple sources of information for work-piece and machine condition monitoring, and equipment time to failure prediction in manufacturing processes, and provides feedback to shop supervisor. Firstly, we propose a method to predict component wear and failure based on operational data. ICP (Interactive Closest Point) algorithm is used to find the best matching tool path given a certain tool number to identify similar machining processes. The result of ICP tool path matching, together with other parameters such as spindle speed, feed rate and tool number, are used to adaptively cluster the machining processes. For each process cluster, a particle filter based prognostic algorithm is used to predict tool wear and/or spindle bearing failure. Secondly, we propose to use anomaly detection methods to detect changes in normal behavior of the machines. Various machine learning algorithms are utilized to detect anomalies based on real-time data, and a voting mechanism is used to decide when to trigger an alarm. Thirdly, the axes traverse is aggregated to provide a measure of the wear on various axes in the machine, which is correlated to errors in position comparing to the commanded positions and nominal tool paths. Spindle load verse rotating speed is also examined to facilitate shop floor scheduling to avoid damage caused by unintentionally excessive machine usage. The proposed framework has been demonstrated using published data from two Mazak machine tools.

Linxia Liao et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Sparked by IT megatrends, manufacturers are currently undergoing an operational transformation with increased agility and efficiency. Key technologies influencing this change include digital manufacturing, cloud computing, mobile application, and big data. At the intersection of these technologies there is an opportunity to create a self-aware machine platform in manufacturing shop floor. With the advancement of sensing technology and automation, more information can be derived to facilitate better collaboration and decision making.

Some of the most critical factors, influencing the output of a machining process, are related to tooling, operating parameters, and the ability of a machine tool to maintain its accuracy and repeatability. Changes due to wear or failure of critical machine tool components can lead to significant losses in production and unexpected downtime. One of the current barriers of condition monitoring systems is that the collected sensor data are not well correlated with the in-process machining operating conditions, which compromises the prediction accuracy. Another barrier is that the typical assumptions underlying the prediction of time to failure algorithms (e.g. exponential fault growth) are rarely applicable in real machining. In addition, existing systems operate independently, and impose proprietary interfaces and machine communication protocols that can lead to excessive time consuming and expensive installations.

The goal of the proposed framework is to develop a self-aware system capable of integrating multiple sources of information for work-piece and machine condition monitoring, and equipment time to failure prediction in manufacturing

processes. Currently, the primary applications developed using MTConnect
(MTConnect, 2009) data are focused on the visualization and reporting of OEE (Overall Equipment Effectiveness) and history of alarms. The proposed method goes beyond reporting to provide insight for cell operators on accumulated damage and use automatic clustering for process grouping with particle filter based prognostics using time series data to provide early warning systems for tool wear. Rigid body registration algorithms are used to automatically identify segments of tool paths that can be used to predict or reinforce tool wear prediction. Multiple anomaly detection algorithms with a voting mechanism are used to detect process anomalies across machines. We believe that machine self-awareness will drive the value chain from traditional fail-and-fix, preventive maintenance, condition based monitoring towards self-adaptive, self-analyzing and coordinated assets (see Figure 1)

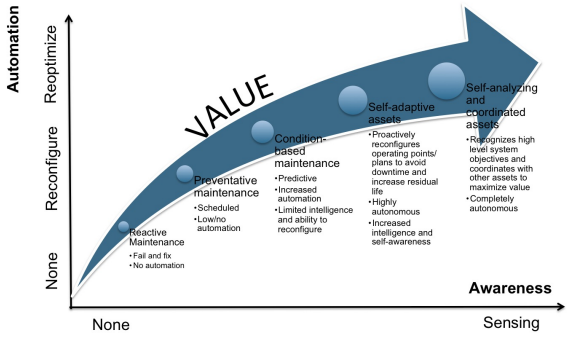


Figure 1. A vision of self-aware machine.

2. THE PROPOSED FRAMEWORK

The proposed framework uses MTConnect data alone to derive information of health condition estimation and prediction for machine components, process anomalies detection across machines using machine learning methods, provide shop floor planning recommendation using statistics.

2.1. Data Collection and Preprocessing

For demonstrating our framework, we use data provided at a public URL for the MTConnect challenge. A query post (e.g. `http://66.42.196.109:5605/sample?count=2000`) is sent periodically to the MTConnect enabled machine IP address. The query returns an XML (Extensible Markup Language) formatted file which contains all the data published from the machine. Since we query periodically, the data returned by a query may contain some data that was also returned as part of a previous query. To avoid data redundancy, we check the sequence numbers returned from the query result to record data when it is updated. Using the tags 'nextSequence', 'firstSequence', and 'lastSequence', we ensure that 'nextSequence' is greater than 'lastSequence' and 'nextSequence' increases by the count number compared to its previous value (e.g. count number is set to 2000 in the query example shown above). A snapshot of the data XML file is shown in Figure 2.

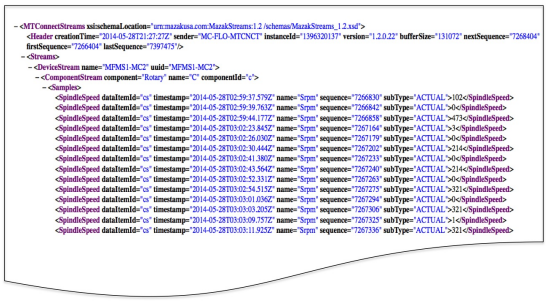


Figure 2. An example of MTConnect data file in XML format.

A parser is written to obtain the time stamps and values of the variables from the tags in the returned data file. The variables that we obtained include x-axis position, y-axis position, z-axis position, spindle load, x-axis load, y-axis load, z-axis load, feed rate, feed rate override, spindle speed, spindle speed override, and tool number. The data is updated when the value of a variable is changed. Hence, for a certain time stamp, there may be no value for a variable because it is not updated at the time stamp. If there is no value available, the previous value is inserted at the time stamp since the value hasn't changed yet. After the parsing and insertion, a vector of a time stamp and the values of all the aforementioned variables are obtained. This allows us to get a matrix of data indexed by multiple time stamps.

2.2. Component Level Health Monitoring and Prediction

One of the characteristics of a self-aware machine is to be able to detect its components degradation and predict future failure. The components (e.g. spindle, cutting tool, and feed axis) on a machine are often used under different machining processes in a manufacturing shop floor. A machining process in our research is defined as a cutting tool with the same tool number sharing similar tool paths with the same non-zero spindle speed and feed rate (overridden value) associated with a certain time period. For each process, the spindle power data were recorded as wear indicators. An adaptive clustering method is applied to cluster the different processes. Prediction is made using a filtering method to predict component failures with data from the specific process as well as data from other processes using the same tool. The prediction provides insight into every single process, which not only guides the maintenance decision makers to take proactive actions on the machine component to avoid unplanned

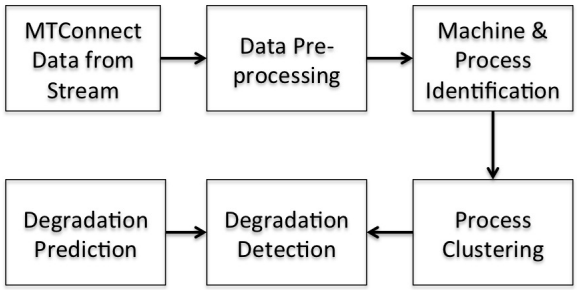


Figure 3. Flowchart of component level health monitoring and prediction.

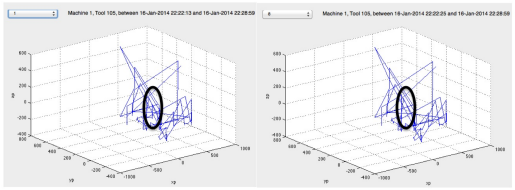


Figure 4. The tool paths of similar machining processes.

downtime, but also assists the process planners to track the production drawbacks to improve their process design. The flowchart is shown in Figure 3.

- Machine and Process Identification
  Different machines are using different IP addresses to publish the data. The identification of the machine will be determined by the IP address used in the query post described in Section 2.1. For a specified cutting tool, the tool path consists of multiple x, y, and z positions. The spindle speed and feed rate change during machining. For the same part, x, y, and z positions determine the shape of the tool path in 3-D space (shape space). The spindle speed, feed rate and time form another 3-D space (parameter space). For two machining processes, if the same cutting tool is used for the entire machining process and the shape space and the parameter space are both matching, we assume these two machining processes are similar processes. The shape spaces of two similar processes are shown in Figure 4. There are small variations in the circled area. This could be happening because the MTConnect protocol has a limitation in the sampling rate. Other than that, the entire tool paths of these two processes are very similar.
  We use ICP (Interactive Closest Point) algorithm (Savoye, 2012) to determine how the shape space and parameter space match. ICP is a commonly used algorithm to align two free-form point clouds in 3-D space. It optimizes the transformation matrices such as scaling, rotation, and translation applied on the target shape to minimize the error with the source shape. It has been successfully used in many fields such as manufacturing (3-D surface inspection), and healthcare (medical image segmentation). We use ICP algorithm to find the best matching machining processes. Let us denote the original 3-D space points cloud as $source$, the transformed points cloud as $tranform$, and the targeted points cloud as $target$. The operation matrix of rotation, scaling and translation are T, b and c, respectively. After the operation we obtain

$$transform = b * source * T + c \tag{1}$$

The ICP algorithm optimizes the operation matrix of T, b and c so that the difference (denoted as d) between $tranform$ and $target$ is minimized. The difference shows the extent to which $source$ and $target$ are different. The smaller the difference, the better the match/overlap between $source$ and $target$. The difference between the shape spaces is denoted as d_s, and the difference between the parameter space is denoted as d_p. The matching measure is denoted as $d_a = [d_s, d_p]$.

- Process Clustering
  Machines are usually programmed to perform different jobs under various machining processes depending on the tasks. To compare the condition of the machine, we need to group the similar processes into a cluster with in which the analysis is performed to derive the health condition. The data stream may contain a brand new process that has not been experienced before. An adaptive clustering method is used to automatically cluster the machining processes into different clusters. If a new machining process is detected (i.e. it does not belong to any existing process clusters), a new process cluster is assigned. If a machining process belongs to an existing cluster, the process is assigned to that cluster and the centroid of the cluster is updated. To determine whether a process belongs to an existing cluster or not, a T2 limit is applied on the matching measure d_a. Let the mean value of the matching measure of an existing cluster be $\hat{d}_a$ and the covariance be s. The T2 statistics for the matching measure of a process is calculated by

$$T2 = (d_a - \hat{d}_a) * s^{-1} * (d_a - \hat{d}_a)' \tag{2}$$

The T2 control limit is calculated by

$$T2_{limit} = \frac{(N-1)(N+1)p}{N(N-p)} F_\alpha(p, N-p) \tag{3}$$

where $F_\alpha(p, N-p)$ is the $100\alpha\%$ confidence level of F-distribution with p and $N-p$ degrees of freedom. If the $T2$ statistic is below the $T2_{limit}$, the process belongs to an existing process cluster; otherwise a new cluster is created for the process.

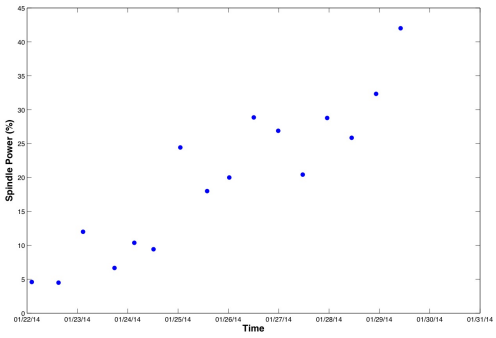


Figure 5. Degradation of cutting tool No. 63.

- Degradation Detection
  After similar processes are grouped into clusters, we can perform degradation detection within each cluster. We assume that the spindle power increase is proportional to the increased severity of tool wear for similar machining processes. The local trend of the power increase may vary (e.g. there may be stochastic variations locally). However, the overall trend of the power should be increasing over time. Hence, a monotonicity criterion is used to detect the increasing trending of the spindle power. Monotonicity is defined in (Coble & Hines, 2009) as:

$$Monotonicity(F) = \frac{\#d/dF > 0}{n-1} - \frac{\#d/dF < 0}{n-1} \quad (4)$$

  where F is the measurement, n is the number of measurement in a period of time. F represents a feature and d/dF is the derivative. The maximum value of $Monotonicity$ equals to 1 only if the feature is monotonically increasing. The value of monotonicity indicates the increasing trend of the spindle power, which indirectly indicates the degradation of the cutting tool. Figure 5 shows the detected trend of the cutting tool number 63.
  This analysis will be performed within all the process clusters. If multiple processes belong to a same cutting tool and degradation trend has been detected with these processes, it is more certain that the cutting tool is wearing.
- Degradation Prediction
  If a degradation trend is detected, we can extrapolate the trend to infer the remaining cuts under the same process given a preset threshold of the power. A particle filter (Chen, Zhang, Vachtsevanos, & Orchard, 2011) can be adapted for the prediction due to its capabilities to cope with system non-linearity and estimate prediction uncertainty. The prediction is made using a continuous Bayesian update method assuming the fault growth following a physics-based system degradation model (e.g. the Paris' Law), which is widely used as the fatigue crack growth model. The system degradation was assumed to be a first-order Markov process, i.e. the current state was only dependent upon the last state. In this case, we observed that the degradation trend was closely following a second order polynomial model such as:

$$X_k = a_k t_k + b_k t_k^2 + c_k \quad (5)$$

  where X_k is the system state (tool wear in this case), t_k is the time at step k, and a_k, b_k, c_k are the parameters of the second order polynomial model. We can write Eq.(5) into the format of a Markov model as follows:

$$\begin{aligned} X_k &= a_k t_k + b_k t_k^2 + c_k \\ &= a_k(t_{k-1} + \Delta t) + b_k(t_{k-1} + \Delta t)^2 + c_k \\ &= a_k t_{k-1} + b_k t_{k-1}^2 + c_k \\ &\quad + a_k \Delta t + 2 b_k t_{k-1} \Delta t + b_k \Delta t^2 \\ &= X_{k-1} + (a_k + 2 b_k t_{k-1}) \Delta t + b_k \Delta t^2 \end{aligned} \quad (6)$$

  The parameter identification and state estimation can be performed in parallel. The prediction (median of the particles) of the remaining cuts for the degradation situation shown in Figure 5 is 13 give 70% of spindle power as the threshold. This information can alert the maintenance team to change the cutting tool before it fails.

2.3. Process Anomaly Detection Across Machines

Anomaly detection (Barnett & Lewis, 1994), (Hodge & Austin, 2004) is an important concept for a self-aware system. An anomaly is simply an exception or deviation from the typical usage (tools, power, speed etc.) and does not necessarily imply a malfunction. For example, machining a new part or using a new tool or working with a new type of material may all be deviations from the previous usage of a machine. However, these are intended (and desired) deviations - on the other hand, if the power usage is unusually high despite unchanged job parameters then it may point to an underlying condition. So a self-aware machine can indicate to the operator that it is experiencing a significant deviation from its typical behavior - the operator can decide whether the deviation is a cause for concern. In fact, the operator can annotate the behavior for future use. So if the anomaly is just a desired new behavior then it can be labeled as such and the machine will know not to flag it in the future. On the other hand, if it is an indication of an underlying condition then it can be labeled with the diagnosis and the machine can flag it appropriately in the future. In this section, we show how anomaly detection can be performed on MTConnect data to identify deviations in usage. While not as informative as the approaches mentioned in Section , anomaly detection can be very scalable as it need not rely on models of failure.

As mentioned in Section 2.3, we analyze data from an MTConnect stream. Let us look at a snippet of this data shown in Table 1. The first six columns provide a time stamp for the

data while the remaining columns provide details about the job (tool ID, feed rate, spindle speed, tool path, and spindle power) - we use the job parameters for our analysis. In the literature, there are a number of popular approaches to anomaly detection. Here, we consider three: 1) self organizing maps (SOMs), 2) regression, and 3) Mahalanobis distance.

2.3.1. Self Organizing Maps (SOMs)

SOMs (Kohonen, 2001) are a natural way to organize an incoming stream of data into a grid of cells - a (typically Euclidean) distance metric is used to assign new data instances to cells containing similar data. As data accumulates, some cells will become very dense and will represent the typical behavior/usage of the machine. If a new data instance is assigned to sparsely populated cell then that would indicate a deviation from the typical behavior/usage. If this behavior is desirable or intended then the cell can be labeled as such. Otherwise, it can indicate undesired behavior or malfunction. For this data, a SOM is shown in Figure 6. While the data is high-dimensional, for ease of visualization we have only shown spindle speed (x-axis) and spindle power (y-axis). We start with a 7x7 grid evenly distributed on the space spanned by the expected range of the variables. Then we assign points to the cells in an incremental manner based on the Euclidean distance. After a data point has been assigned, the cells are warped to have a greater resolution in areas of high density (i.e. areas representing usual behavior) - please see (Rougier, Boniface, & Universit, 2011) for more details. The gray lines in Figure 6 represent the Voronoi partition (`http://en.wikipedia.org/wiki/Voronoi_diagram`) of this grid where each partition represents the extent of the corresponding node - a data point within a partition is assigned to the node associated with it. Due to the warping, the structure of the data clearly stands out. The lower left corner has small and dense cells representing the typical usage of the machine. The space of large spindle speeds and power is very sparse. There is a clear anomaly in the top right corner corresponding to spindle power of 87 units and spindle speed of 3127 rpm - in addition, there are many sparse cells corresponding to higher than usual values of speed and power. If a new data point falls in a sparse or hitherto unseen region, it can be flagged for review. The operator can choose to investigate and annotate the cell for future reference.

2.3.2. Multivariate Regression

Another way to look at this problem of self-awareness is from the perspective of relationships between the variables. In a control system such as a CNC machine, the high level requirements (e.g. the tool path) are translated into low level specifications (e.g. feed rate, spindle speed etc.) which are then met using control inputs (e.g. spindle power). So it may be quite normal for power usage to be high if the required speed is high. If we can learn the normal relationship between the different variables then it should be possible to raise a flag when the variables of a new data instance exhibit a significantly different relationship. In this section, we show how multivariate regression may be used to learn the relationship between variables.

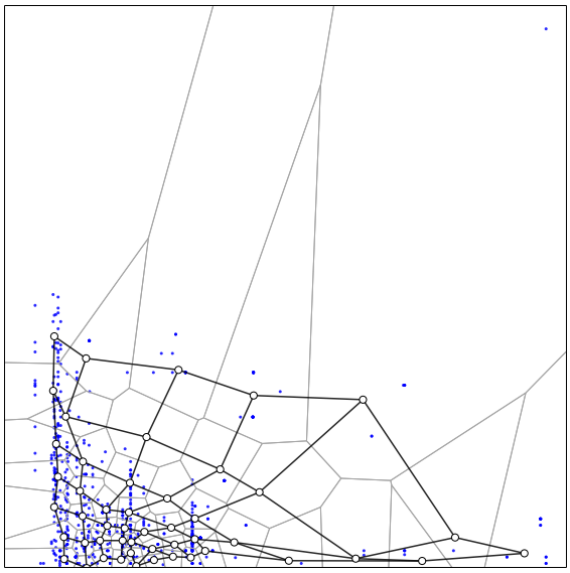


Figure 6. A Self-Organizing Map for MTConnect Data from a Mazak Machine

Table 2. Processed MTConnect Data

tool ID	duration	spindle speed	feed rate	distance	spindle power
0	0.083	400	1.19	0.81	13
0	0.70	1131	26.84	194.82	7
...	...	...	...	...	...

Before performing regression, we need to pre-process the data. In Section 2.3, we mentioned that ICP path matching as a approach for analyzing the tool path - it ensures that the analysis performed is invariant with respect to affine transformations of the tool path. The primitive for our regression analysis is not the entire tool path but rather the sampling interval of the data collection process - executing the entire tool path may take many minutes but the data being analyzed is sampled every few seconds. So rather than analyzing the entire tool path, we analyze the distance traveled by the tool during a sampling instance. This is just a design choice - domain expertise can be used to pick a different primitive. After pre-processing, we get data of the following form:

Here *tool ID* is a categorical variable[1] while the others are real numbers - we try to learn a model to predict spindle power based on the other variables. There are many modeling ap-

[1]There are 36 distinct tool IDs: 0, 10, 102, 104, 107, 108, 109, 111, 112, 115, 117, 118, 120, 17, 2, 20, 24, 25, 3, 32, 4, 44, 45, 5, 52, 58, 63, 65, 69, 70, 74, 77, 88, 90, 92, 98

Table 1. MTConnect Data

year	month	day	hour	minute	second	tool ID	feed rate	spindle speed	x	y	z	spindle power
2014	1	23	14	51	28	0	1.19	400	2.11	-32.46	-70	13
2014	1	23	14	51	33	0	1.19	400	0	-32.46	-69.14	13
...	...	...	...	...	...	...	...	...	...	...	...	...

proaches for regression but we are specifically interested in two characteristics: 1) ability to provide a prediction interval for new data points, and 2) ability to build accurate models without making assumptions about the nature of relationship between the variables. The first requirement (prediction interval estimation) is necessary for defining anomalies (deviations) in a structured manner but the second requirement (assumption-free modeling) is just a convenience to enable automation. There are many options but quantile regression forests (Meinshausen, 2006) are ideally suited for this scenario and that is what we used for this analysis. They provide a reasonable fit to the data and give us the ability to estimate prediction intervals based on user defined quantiles. Let Q_α be defined as

$$Q_\alpha(x) = \inf\{P(Y \leq y | X = x) \geq \alpha\} \tag{7}$$

Then Q_α represents the $\alpha-$quantile for the conditional distribution of a variable Y conditioned on a vector variable X. If Y is the variable being predicted (spindle power in our example) then Q_α defines its $\alpha-$quantile conditioned on the prediction variables X (tool ID, duration, spindle speed, feed rate, and distance in our example). For this analysis, we use $[Q_{0.025}, Q_{0.975}]$ as the prediction interval and designate a new data instance as anomalous if the actual spindle power lies outside the prediction interval. Compared to the SOM approach, this approach has the advantage that we explicitly model the relationship between spindle power (dependent variable) and the other variables (independent variables). The notion of prediciton interval is also a big advantage as it provides a systematic approach to detecting outliers. The prediciton interval will be small if we have a high confidence in our prediction so even small unexpected deviations outside the prediction interval may be flagged. On the other hand, it has the disadvantage that we can only flag anomalies in the value of the independent variable conditioned on the independent variables - we cannot flag anomalies in the independent variables themselves (since they are considered inputs into the model). Typically, excessive deviations in the control signal are good indicators of underlying conditions so this is not a big drawback.

For this dataset, the quantile regression forest achieves reasonable accuracy in predicting the spindle power ($R^2 = 0.74$). However, we are not interested in the actual predictions per se but rather in large errors in those predictions (i.e. values that lie outside $[Q_{0.025}, Q_{0.975}]$. The graph in Figure 7 shows

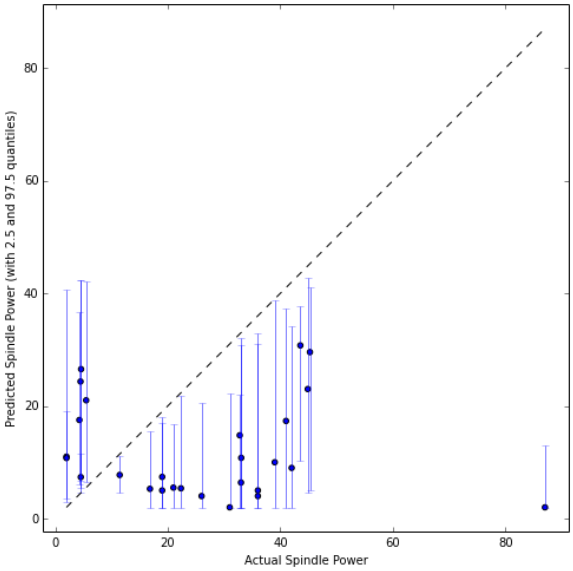


Figure 7. Outlier Detection using Quantile Regression Forest

such deviations. As in the case of SOMs, the instance where the spindle power is 87 stands out as a clear outlier. Most of the other outliers are cases where the actual value lies just outside the prediction interval.

2.3.3. Robust Mahalanobis Distance

If the data are assumed to be samples from a multivariate normal distribution then Mahalanobis distance can be used to detect outliers. In that case, outliers are data points that are samples from a different distribution rather than extreme values of the multivariate normal distribution. This has the advantage that we don't need to choose a cutoff point for labeling a point as outlier - we simply look for points that likely came from a different distribution (see (Filzmoser, Garrett, & Reimann, 2005) for more details). Of course, the normality assumption may not be satisfied in reality - in fact, it is not satisfied for the data set being used here. In that case, we can still use Mahalanobis distance to look for outliers without relying on distributional assumptions. One approach is to transform the data into the principal component space and look for the outliers in the space spanned by the top few principal components. Since principal components are aligned with directions of maximal variance, that makes it easier to spot the outliers. Also, by looking in the reduced space of the top principal components, it increases the signal to noise ratio. Using ap-

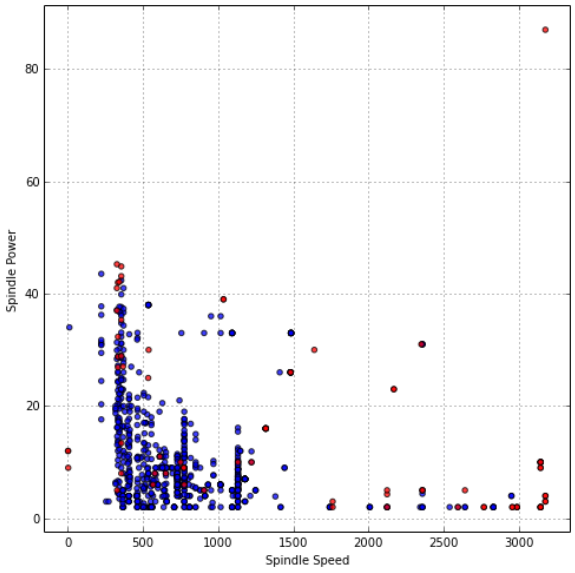


Figure 8. Mahalanobis Distance Based Outlier Detection

propriate normalization (see (Filzmoser, Maronna, & Werner, 2008) for more details), the Euclidean distance in the principal component space is equivalent to Mahalanobis distance in the original space. In the absence of any distributional assumptions, (Filzmoser et al., 2008) proposes a measure of *outlyingness* of a data instance based on its Mahalanobis distance. We use that same measure in our analysis here.

The results are show in Figure 8 - the outliers are shown in red[2]. The instance where spindle power is 87 is again identified as a clear outlier in addition to some others.

2.3.4. Ensemble of Outlier Detection Methods

In this section, we discussed three outlier detection approaches, namely, self-organizing maps, multivariate regression, and robust Mahalanobis distance. There are many other other methods that could be applied. All these methods make different assumptions and have different strengths and weaknesses. We can combine them into an ensemble that can raise flags based on some predetermined policy. For example, if the cost of failure is very high then the ensemble may flag a data instance as an outlier if any member of the ensemble determines the data instance to be an outlier (this would be an *OR* policy). Alternatively, if the cost of disruption of workflow outweighs the cost of failure then the ensemble may flag a data instance as an outlier only if all members of the ensemble agree (this would be an *AND* policy). In most scenarios, a good policy might be for the ensemble to flag a data instance as an outlier if a large fraction of the ensemble members agree (this would be a *MAJORITY* policy).

[2]This multivariate analysis included duration, feed rate, spindle speed, distance, and spindle power but we only show the spindle speed and power in the graph for ease of visualization.

2.4. Shop Floor Planning Recommendation

Another aspect of machine self-awareness is that the machines are able to compare their usage and performance with each other. The information can be fed back to the shop floor planning trying to avoid damage due to unintentionally excessive usage by rescheduling the machining tasks.

The spindle data can be used to estimate spindle damage as the bearing life is proportional to $load^3 * rpm$ (revolutions per minute). The aggregate axes traverse provides a measure of the wear on various axes in the machine (an estimate of the way damage). This can be correlated to error in position if either commanded position is available via MTConnect protocol or nominal tool paths are available to switch the axis to condition based maintenance. This recommendation provides insights by shop defined rules ifor switching parts between machines if any axis travels beyond a threshold greater than twice that of a comparable machine in the same time frame.

Figure 9 contains an overview about a cell of machines. The machines are identified by the individual MTConnect Stream. We use the data from two machine provided by MTConnect challenge (`http://66.42.196.109:5605/current` and `http://66.42.196.109:5606/current`). The figure has three distinct sets of information presented: recommendations for the cell based on data, histogram plot of spindle rpm (revolution per minute) weighted by the load at the specific rpm, and total traverse compared across different feed axes on the machine. MTConnect provides insight into usage of machines both absolute and relative to each other in a cell when aggregated over time. The histogram of the spindle loads weighted by the time spent at various spindle speeds provide a relative estimate of remaining useful life (RUL) of the spindle bearings. This information can be fed back to the scheduling systems depending on the shop's maintenance policy. For example, if all machines will be taken down around the same time for service, this can be used to balance the spindle loads across machine. Similar analysis can be employed to balance travel of various drive axes by shifting parts appropriately. These include rotating the fixtures based on current state and scheduled tool paths.

This helps shop supervisors balance usage across machines at a deeper level than utilization to reduce excessive damage accumulation on a single machine in a cell while reducing unexpected downtime for individual machines. The recommendation will enable manufacturing shops to move from scheduled maintenance to condition based maintenance based on true damage accumulation.

3. CONCLUSION AND DISCUSSION

The framework we have developed is scalable with broad applicability for milling, drilling, turning machines in various configurations. It can be configured from cell level to

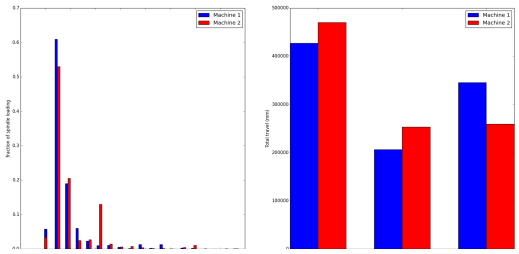


Figure 9. Shop floor recommendation for spindle and axis planning.

plant level with minimal effort and is applicable for small and medium-sized or large enterprises. It also has broad based applicability for various industries including fabricating industrial components, such as automotive engine, medical device, or aerospace parts. Only part of the MTConnect data is considered in our research. More variables can be used to obtain the machine health information from a broader view. The sampling rate has certain limitations as mentioned in the previous section. More information can be derived by combining operational data with external sensor data (e.g. vibration, acoustics signal) to gain more insight about the machine component health, e.g. (Liao & Pavel, 2012) and (Liao, Edmondson, & Ludwig, 2012).

Machine self-awareness could shift the industry from a reliance on a preventative paradigm (checking performance and replacing parts on a set schedule, regardless of whether there is an immediate need for these activities), to a predictive paradigm (schedule maintenance before failure actually happens). Self-aware machines will positively impact production time, cost, and quality of any manufacturing plant by reducing unplanned downtimes, adapting for work-piece variability, and enabling specification of fault-tolerant process plans.

REFERENCES

Barnett, V., & Lewis, T. (Eds.). (1994). *Outliers in statistical data*. Wiley New York.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard, M. (2011). Machine condition prediction based on adaptive neuro–fuzzy and high-order particle filtering. *IEEE Transactions on Industrial Electronics*, *58*(9), 4353-4364.

Coble, J., & Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Proceedings of the annual conference of the prognostics and health management society.*

Filzmoser, P., Garrett, R. G., & Reimann, C. (2005). Multivariate outlier detection in exploration geochemistry. *Computers & Geosciences*, *31*(5), 579-587.

Filzmoser, P., Maronna, R., & Werner, M. (2008). Outlier identification in high dimensions. *Computational Statistics & Data Analysis*, *52*(3), 1694-1711.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85-126.

Kohonen, T. (Ed.). (2001). *Self-organizing maps*. Springer.

Liao, L., Edmondson, Z., & Ludwig, H. (2012). Plug and prognose - condition monitoring, diagnosis and life time prediction. *ATP edition*, *54*(10), 52-56.

Liao, L., & Pavel, R. (2012). Machine tool feed axis health monitoring using plug-and-prognose technology. In *Proceedings of the 2012 conference of the society for machinery failure prevention technology.*

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*, 983-999.

MTConnect. (2009). *Mtconnect standard part 1-overview and protocol, version 1.01* (Tech. Rep.). MTConnect Institute.

Rougier, N., Boniface, Y., & Universit, L. (2011). Dynamic self-organising map. *Neurocomputing*, *11*(74), 1840-1847.

Savoye, Y. (2012). Iterative cage-based registration for dynamic shape capture. In *Acm siggraph 2012 posters.*

BIOGRAPHIES

Linxia Liao Linxia Liao's research interests include predictive and prescriptive analytics; system and components fault diagnostics and prognostics; signal processing; and machine learning algorithms as well as their integration on embedded systems. Currently, he is developing device fleet health management solutions for intelligent transportation systems. Prior to joining PARC, Linxia worked as a research scientist with Siemens Corporation, Corporate Technology (previous Siemens Corporate Research) located in Princeton, NJ. He conducted research and implemented various prognostics and health management-related applications in the fields of manufacturing, energy, and transportation. He also previously worked at Siemens Technology-To-Business (TTB) Center in Berkeley, CA to transfer the patented 'Methods for prognosing mechanical systems' technology from the university to industry applications. Linxia received his Ph.D. degree in Industrial Engineering from the University of Cincinnati, where he conducted research at the NSF I/UCR Center for Intelligent Maintenance Systems (IMS). Linxia has one issued patent and seven pending patents, and he has published one book chapter and 20+ papers in leading journals and conferences.

Raj Minhas Raj Minhas is the Program Manager for Prognostics and Health Management and is responsible for the strategy and execution for the commercialization of the related technologies. Prior to PARC, he was the Director of Xerox Research Center India where he led its growth, development, and outreach for two years. He held a variety of leadership roles at Xerox Research Center Webster includ-

Fault Tolerant Control for Manufacturing Discrete Systems by Filter and Diagnoser Interactions

A. Philippot[1], P. Marangé[2], F. Gellot[1], J.F. Pétin[2] and B. Riera[1]

[1]*Centre de Recherche en STIC (CReSTIC) – University of Reims Champagne-Ardenne (URCA), Reims, France*
{alexandre.philippot, francois.gellot, bernard.riera}@univ-reims.fr

[2]*Centre de Recherche en Automatique de Nancy (CRAN) – University of Lorraine, CNRS, Vandœuvre-lès-Nancy, France*
{pascale.marange, jean-francois.petin}@univ-lorraine.fr

ABSTRACT

The paper deals with an online safety mechanism to define interactions between a diagnoser and a control filter for fault tolerant control of manufacturing discrete systems. The diagnoser observes the plant behavior whereas the control filter ensures the safety from the controller. This online interaction is based by events communication where the control law is never reconfigured. The proposed approach is applied to CISPI platform from the CRAN laboratory (Research Center for Automatic Control of Nancy).

1. INTRODUCTION

Engineering systems become more and more complex and consequently, faults are more and more present and cause undesired behaviors. Diagnosis information can lead the user in its decision for maintenance or reconfiguration (Nke and Lunze, 2011), but can also allow fault tolerant control. The aim of diagnosis approaches is to detected and isolated with certainty a fault. After this step, it is necessary to reconfigure the controller in order to guarantee the dependability and safety but also to propose a Fault Tolerant Control (FTC) in a degraded mode (Blanke et al., 2003, (Paoli et al., 2011, Brown and Vachtsevanos, 2011).

Ensuring safety of manufacturing system control is currently based on two complementary approaches: control design activities with the objective to avoid unexpected behaviors and safe design activities by the development of online barriers.

First one, we focus on the control design activities with the objective to avoid unexpected behavior. Two main approaches are suggested in this way (Faure and Lesage, 2001): (i) control validation and verification (V&V) (Roussel and Faure, 2002), (ii) Supervisory Control Theory (SCT) based on synthesis controller (Ramadge and Wonham, 1989), that enables automatic generation of the controller from the specification, and the uncontrolled behavior of the plant. Most of the time, those designing approaches make two strong assumptions: the behavior of plant devices is not faulty and the designed control is exactly the same as the program that is implemented on the control devices (i.e. code generation deviations or code modifications by maintenance agents are not considered).

These assumptions being not realistic in practice, a second approach complements the safe design activities by the development of online barriers like diagnosis or filtering control. Diagnosis of manufacturing systems aims at detecting unsafe behavior of the plant and localizing the components that are involved in the behavioral deviation (Sampath, 1995). Control filtering aims at avoiding that a PLC program provokes plant damages, whatever the PLC program (Marangé, 2008, Riera et al., 2012). The filter is placed between the controller and the plant and inhibits potential dangerous evolutions by checking a set of safety constraints. Nevertheless, the diagnosis and the filter are formally built from models of process behavior. Consequently, hypothesis that the information from the process is correct is made. At least, if the plant situation is unknown, automatic procedures implemented by control filtering and diagnosis may be not efficient. This case generally requires the intervention of human expert to analyze the unknown situation of the plant, and to take emergency decision to drive back the plant in acceptable states.

The aim of this paper is to propose an approach of FTC where diagnosis provides information about the plant to the filter; and vice-versa. Control laws are never reconfigured but the system must always be in safety situation thanks to the filter even in case of plant fault. Models of the plant devices behavior as well as the control rules can be described as Discrete Event Systems (DES), i.e., dynamical systems with discrete state spaces and event-driven

Alexandre Phhilippot et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

transitions (Cassandras and Lafortune, 1999). The proposed approach provides similar results in term of detection to classical approaches (Sampath, 1995, Debouk, et al., 2000, Wang et al., 2007 …) but it continues to improve the safety even in presence of faults thanks to the control filter.

The paper is organized as follows. In section 2, the fault tolerant control architecture proposed is presented with a diagnosis and a filtering control sub-sections. A benchmark is studied with results in section 3 before to conclude and propose some future works.

2. FTC ARCHITECTURE

From the previous discussion, diagnosis approaches make hypothesis that controller information is safe whereas filtering controller approaches are supposed free of faults. The figure 1 presents the FTC architecture. Control law, diagnoser and filter are present in a Remote Terminal Unit (RTU) as a Programmable Logic Controller (PLC) for example. The diagnoser does not use directly the orders sent by the controller but the orders validated by the filter, which set to allows to guarantee the orders correctness. Also, the filter confirms orders according to the plant information (value of sensors/actuators) and the plant state defined by the diagnoser. User can send requests but also have situation awareness thanks to filter and diagnoser.

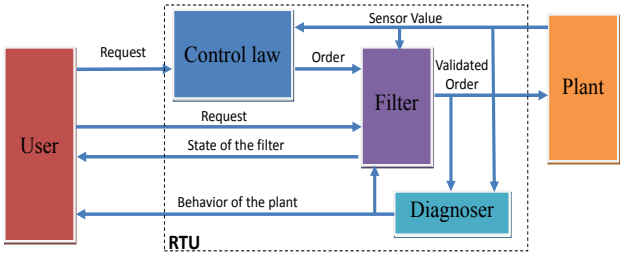


Figure 1. FTC Architecture

2.1. Diagnoser

In industrial processes, a manufacturing system is a functional chain composed of a controller that emits signals to a plant and receives sensor values. This exchange between controller and plant represents the only observable information available online. Since a diagnoser is defined as an observer of the system, it is necessary to use this information to rebuild behaviors through models.

From literature (Sampath, 1995, Qiu, 2005), centralized approaches appear as unthinkable for large and complex systems. As manufacturing system is composed of mechanical components (actuators/sensors), a methodology to obtain a decentralized diagnosis approach, as (Debouk, et al., 2000, Wang et al., 2007, Kan et al., 2010), for manufacturing systems with discrete sensors and actuators has been developed in previous works (Philippot and Carré-Ménétrier, 2011). It is composed of 4 offline steps describe:

1. From the plant components, decomposition is made to obtain local models called Plant Elements (PEs). A PE describes all possible mechanical evolution of the component independently of the controller.
2. From each PE, local desired behavior is extracted. Temporal information, obtained by excited events simulation, is added to enrich the model. The result is an automaton called Normal Behavior Model (NBM).
3. The third step identifies, from each normal state of NBMs, faults which can occur and composes the abnormal model by adding of labeled states to obtain local diagnosers (D_i). Faults are grouped according to the failing component (sensor/actuator) into partitions.
4. A High Level Diagnoser from global specifications is done for uncertainty cases.

Diagnosers are implemented as online observers in the PLC. User's decision is given thanks to the set of local labels.

A local diagnoser is a special case of an observer that carries fault information by means of labels attached to states. These labels indicate the types of faults that have been occurred. A local diagnoser is considered as an extended automaton: $D_i = (X_i \cup X_{DFi}, Z_{io}, \delta_i, x_{i0}, T_i, l_i)$ where:

- X_i is the set of normal states of NBM_i,
- X_{DFi} is the set of faulty states,
- Z_{io} is the set of observable events by the PE_i,
- δ_i: $X_i \times Z_i^* \rightarrow X_i \cup X_{DFi}$ is the transition function with the expected (δ_{ei}) and unexpected (δ_{ui}) functions from a state,
- x_{i0} is the initial state,
- T_i is the set of interval time where transition functions are expected between $[t_{min}, t_{max}]$,
- l_i is the set of decision functions of the local diagnoser D_i with $l_i(x)$ the decision function of the state x which can be one or more fault labels {Fj}. The sets of failure events corresponding to partitions, noted Π_f.

Indeed, the methodology is dependent of the control specification (step 2) and if the controller is not safe or if it changes, then diagnosers can return a bad decision in the first case or must be reconstructed in the second case. To have diagnosis independent from the control, diagnoser is obtained from the behavior of PE and the addition of the possible faulty events.

From decentralized diagnosers, a transition function δ_i corresponds to a logical expression composed by all the events. It is possible to define all transition functions by the 2^n possibility (with n: number of events and intervals). However, the mechanical structure of components and the use of filters make it impossible some combinations. For example, only one interval time can be activate simultaneously, or thanks to the control filter, opposite orders cannot be sent. Consequently, the complexity

depends on the granularity of the local models but also on the performance of the control filter. These diagnosers are independent of the controller specification in its structure thanks to the control filter but not in the definition of the set of interval T_i.

The choice of an automaton to represent a local diagnoser permits to compose a library of commonly components. However, this model can be translated as Markov chain or Causal Temporal Signature under some hypothesis.

2.2. Control Filter

The control filtering consists in interlacing a filter between the plant and the control law to inhibit the evolutions that can lead the system to a dangerous situation for operators and production resources. This aim is to ensure that the controller outputs (Σ_c), are legal according to plant safety. It means that, for each new evolution of actuators output vector (at *t*), the filter verifies that these outputs are compatible with the plant state perceived by means of uncontrollable variables Σ_{uc} (inputs sensors (at *t, t-1, t-2*…), previous outputs (at *t-1, t-2*…), observers (at *t, t-1, t-2*…)).

The filter is built according to a set of logical constraints that must be satisfied to let the outputs getting out of the control filter. It is based on the use of safety constraints, which act as logical guards placed at the end of the PLC program, and forbids sending unsafe controllable events to the plant (Marangé et al. 2008), (Riera et al., 2014). Constraints (or guards) are always modeled with the point of view of the control part (PLC), and it is assumed that the PLC scan time is sufficient to detect any changes of the input vector (synchronous operation, possible simultaneous changes of state of PLC inputs).

Safety constraints are expressed in the form of a logical monomial function (product of logical variables, as $\prod$) which must always be equal to 0 (FALSE) at each PLC scan time in order to guarantee the safety. It is considered in this work that the initial safe state for all the actuators (o_k) is defined to 0.

Initially, the constraints are defined in order to ensure a permissive control, and it is assumed that, with the filter, the system remains controllable. In other words, it is possible to design a controller which matches the specifications. For example, considering the previous hypothesis about the safe initial state, a filter which resets all outputs is safe but does not ensure the controllability. Some guards involve a single output at time *t* (simple safety constraints *CSs*), other constraints involve several outputs at time *t* (combined safety constraints *CSc*). Constraints require the knowledge of Σ_c and Σ_{uc} at the current time *t* and possibly previous times (presence of edge (*t-1*) for instance noted *). Hence, the filter requires a memory function.

The set of constraints CS is considered as necessary and sufficient to guarantee the safety. In this approach, it is assumed that safety constraints can always be represented as a monomial and depend on the uncontrollable and controllable variables (at *t, t-1, t-2*…). Filter stops has to stop the process in a safe situation if a safety constraint is not respected.

CSs and *CSc* can be represented respectively by equation (1) and equation (2) which are Boolean monomial functions and have always to be False at each PLC scan time. N_{CSs} and N_{CSc} are respectively the number of simple safety constraints and the number of combined safety constraints. N_o is the number of outputs.

$$\forall m \in [1, \mathrm{N_{CSs}}],\ \exists! k \in [1, \mathrm{N_O}]\ /\ CSs_m = \prod(o_k, \Sigma_{uc}) = 0 \quad (1)$$

$$\forall n \in [1, \mathrm{N_{CSc}}],\ \exists! (k, l, \dots) \in [1, \mathrm{N_O}]\ with\ k \neq l \neq \cdots\ /\ CSc_n = \prod(o_k, o_l, \dots, \Sigma_{uc}) = 0 \quad (2)$$

There are 2 forms of Simple Safety Constraints *CSs* because they are expressed as a monomial function, and they only involve a single output at time *t* (equation (3) or (4)):

$$\forall m \in [1, \mathrm{N_{CSs}}],\quad \exists! k \in [1, \mathrm{N_O}]\ /\ CSs_m = o_k . h_{0m}(\Sigma_{uc}) \quad (3)$$

xor

$$CSs_m = \overline{o_k} . h_{1m}(\Sigma_{uc}) \quad (4)$$

These simple safety constraints (*CSs*) express the fact that if $h_{0m}(\Sigma_{uc})$ which is a monomial (product) function of only uncontrollable variables at *t*, is TRUE, o_k must be necessarily FALSE (equation (3)) in order to keep the constraints equal to 0. If $h_{1m}(\Sigma_{uc})$ is TRUE, o_k must be necessarily TRUE (equation (4)).

For each output, it is possible to write equation (5) corresponding to a logical OR of all simple safety constraints.

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(f_{sk}(o_k, \Sigma_{uc}) \right) = 0 \quad (5)$$

$f_{sk}(o_k, \Sigma_{uc})$ is a logical $\sum\prod$ function independent of the other outputs at *t* because only *CSs* are considered. $f_{sk}(o_k, \Sigma_{uc})$ can be developed in equation (6) where f_{s0k} and f_{s1k} are polynomial functions (sum of products, $\sum\prod$) of uncontrollable variables. Equation (6) has always to be FALSE because all simple safety constraints must be FALSE at each PLC scan time.

$$f_{sk}(o_k, \Sigma_{uc}) = o_k . f_{s0k}(\Sigma_{uc}) + \overline{o_k} . f_{s1k}(\Sigma_{uc}) = 0 \quad (6)$$

Taking into account all *CSs*; it is possible to write equation (7).

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(o_k . f_{s0k}(\Sigma_{uc}) + \overline{o_k} . f_{s1k}(\mathrm{I}, \Sigma_{uc}) \right) = 0 \quad (7)$$

The definition of constraints set is not formal and the filter robustness must be verified. In (Marangé, 2008) and (Riera et al., 2012), authors proposed to enrich this expert-based approach by a formal identification of the constraints set to ensure its completeness.

The use of this filter allows detecting errors resulting from the controller by making a hypothesis on the accuracy of the information resulting from the plant. Indeed, a fault on the plant can lead:

- Too much restriction: sensor information is going to be blocked in the most critical state and the constraint is not verified while the plant is not in a critical situation.
- Too much tolerant: sensor information is going to be in the state which verifies all the time the constraint and thus the filter is going to allow to pass dangerous orders for the plant. This case is to be avoided.

The consideration of diagnosis information allows to use the filter in degraded mode. For that purpose, the information resulting from the plant is added by taking into account diagnoser. When a failure arises on a sensor or an actuator, the filter constraints that contain the logical variables associated to the faulty devices becomes unreliable. Authorized signals may be forbidden, and, worse forbidden signals may be authorized. Consequently, the filter constraints must consider the occurrence of a fault or not.

For every fault partition, a flag is set to true when the diagnoser reaches a faulty decision state. This flag determines if the considered variable can be used into the filter constraint (flag=0), or if an equivalent reconstructed information must be used (flag=1). Only the sensor information can be reconstituted by using:

- the expert knowledge (timed or temporal model),
- redundant information or reconstruction logics.

The property defining the dangerous situation has been verified using a model-checker meaning that the filter delivers correct inhibition and authorization even in presence of device faults (with the assumption that the diagnoser is able to detect and localize the fault).

Moreover, as the control filter only concerns safety part and not the functional part, if the component is exchanged or replaced, only the set of constraints corresponding to this component must evolve. For industrial systems, establishment of a constraints library is feasible. In fact, constraints sets are defined for a sub-system of component interaction.

3. CASE STUDY

The approach is applied to the CISPI platform from the CRAN laboratory (figure 2). This platform implements hydraulic processes involving valves, pumps and tanks and various transmitters (flow, pressure…). Local controllers implement basic control loops and are involved in a global mode management control that enables concurrent access to devices for start, shutdown and normal operation procedures. To avoid damages and failures of the system, as well as the human operator's errors, this experimental platform promotes new forms of control organization that exploits the capacity ambient technologies (sensor network, PDA, mobile control...) to favor safe human/system interactions in any place, at any instant and for any plant operation.

Within the framework of this project, the control filter and diagnoser are implanted to bring a help during the supervisory control of the CISPI system. To illustrate the approach presented in this paper, an automatic valve is considerate. This valve can be closed or open by respectively *C* and *O* boolean signals, and two sensors for the open position (*fso*) and for the closed position (*fsc*) are present.

Independently of the control laws, the sub-system valve must always be in a safety mode. For this, an assumption is made that when a fault is on an actuator, all outputs must be deactivated by the filter. If a fault is on a sensor, the sub-system can be tolerant to this fault.

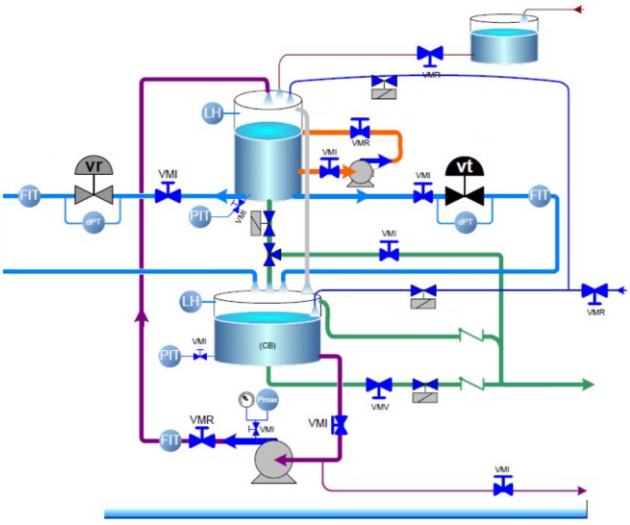


Figure 2. CISPI Platform

3.1. Diagnoser

From the illustrative example, the valve with sensors *fsc* and *fso* constitute one PE and it is possible to identify each faulty event by a label:

- Sensor *fsc* stuck to 0 (F1) or to 1 (F2)
- Sensor *fso* stuck to 0 (F3) or to 1 (F4)
- Valve stuck to *fsc* (F5) or *fso* (F6) position
- Unexpected *fsc* (F7) or *fso* (F9) from 0 to 1
- Unexpected *fsc* (F8) or *fso* (F10) from 1 to 0
- Unexpected movement from *fsc* to *fso* (F11) or from *fso* to *fsc* (F12)
- Valve blocked between *fsc* and *fso* (F13)

Three fault partitions are defined belong to:

- Sensor *fsc*: Π_{fsc} = {F1, F2, F7, F8}
- Sensor *fso*: Π_{fso} = {F3, F4, F9, F10}
- Valve: Π_{Va} = {F5, F6, F11, F12, F13}

With the consideration of the controller information, and thanks to the filter, the valve diagnoser is composed of 9 normal states and 16 abnormal states (Fig. 3) where:

- double circle is the initial state,
- 9 white states are the normal states,
- 3 grey states noted F2, F7, F8 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{fsc},
- 3 grey states noted F4, F9, F10 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{fso},
- 4 grey states noted F5, F6, F11, F13 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{Va},
- 6 black states describe the detection of a fault but not the isolation (4 intermediate before isolation).

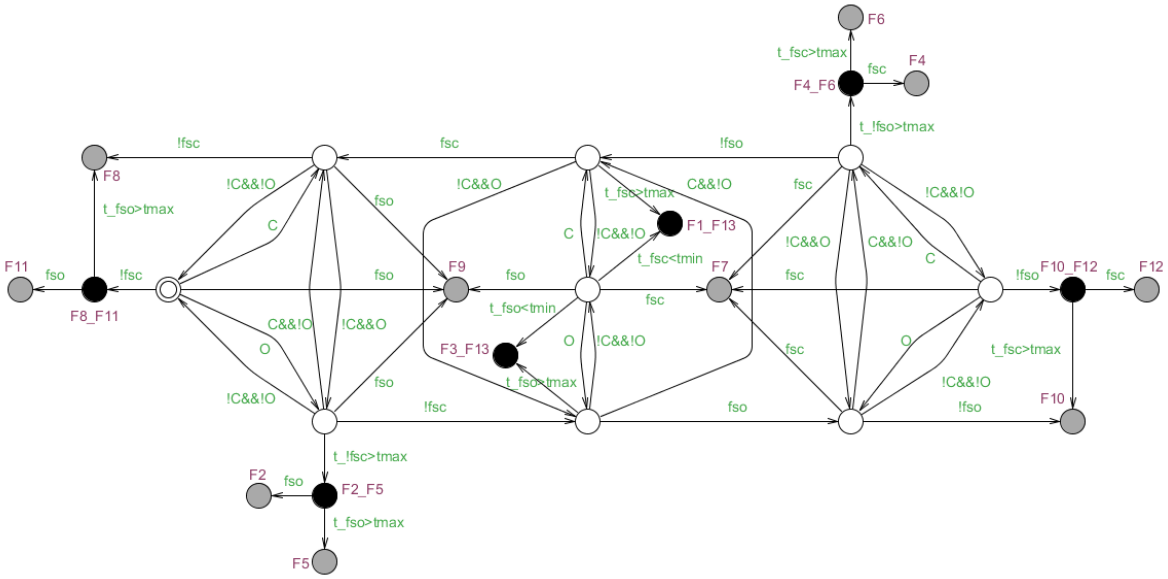


Figure 3. Valve Diagnoser

The reliability of sensors ensures to be into a safety mode (white states). However, after the detection and isolation of a fault (grey and black states), this diagnoser cannot be anew used. Indeed, it is not possible to rely on misinformation. That is why, it is necessary to preserve the state of the system until the fault is been corrected and reset.

3.2. Control Filter

Constraints take into account information of the diagnosers. Information used in the filter is noted X_{filter} and diagnosis information is noted *defX*. The following flags are done:

- *deffso* for the partition of valve sensor *fso*,
- *deffsc* for the partition of valve sensors *fsc*,
- *defV* for the partition of valve actuator *V*,

To be tolerant on sensors' faults, an expert knowledge is used to estimate the plant information by temporal information. This knowledge can be optimally obtained by FMEA (Failure Mode and Effects Analysis) and so provide a reactivity of detection. For example, figure 4 shows equivalent information of *fso* and *fsc* sensors information from a learning chronogram where the estimated value of *fso* is given by a flag *TON1* when an On Delay Timer is activated, and respectively a flag *TON2* for the estimated value of *fsc*.

- for $fso = 1$ by $\widehat{fso} = TON1$
- for $fsc = 1$ by $\widehat{fsc} = TON2$

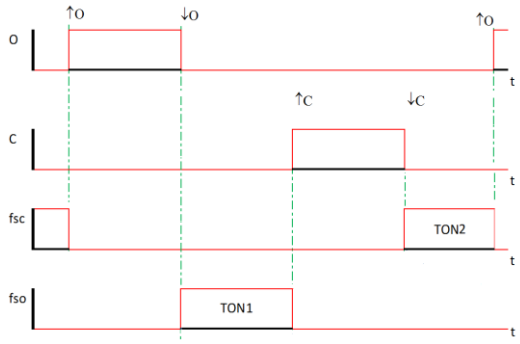


Figure 4. Reconstruction of sensors' information

For a sensor fault, the plant information is replaced by temporal information:

$$fso_{filter} = \overline{deffso}\,.\,fso + deffso.\,\widehat{fso} \qquad (7)$$

$$fsc_{filter} = \overline{deffsc}\,.\,fsc + deffsc.\,\widehat{fsc} \qquad (8)$$

No information can be estimated for outputs *C* and *O*. Consequently, orders must be deactivated by the filter even in case of faulty event by:

$$C_{filter} = \overline{defV}.C \quad (9)$$
$$O_{filter} = \overline{defV}.O \quad (10)$$

The set of constraints is defined as following. It is forbidden to maintain an order when the position valve is done (equations (11) & (12)). It is forbidden to deactivate an order until the ending position valve is not done (equations (13) & (14)). It is forbidden to activate an order until the starting position valve is not done (equations (15) & (16)). For the combined safety constraint of equation (17), it is forbidden to activate orders *C* and *O* together:

$$CSs_1 = C_{filter}.fsc_{filter} = 0 \quad (11)$$
$$CSs_2 = O_{filter}.fso_{filter} = 0 \quad (12)$$
$$CSs_3 = C^*_{filter}.\overline{C_{filter}}.\overline{fsc_{filter}} = 0 \quad (13)$$
$$CSs_4 = O^*_{filter}.\overline{O_{filter}}.\overline{fso_{filter}} = 0 \quad (14)$$
$$CSs_5 = \overline{C^*_{filter}}.C_{filter}.\overline{fso_{filter}} = 0 \quad (15)$$
$$CSs_6 = \overline{O^*_{filter}}.O_{filter}.\overline{fsc_{filter}} = 0 \quad (16)$$
$$CSc_1 = C_{filter}.O_{filter} = 0 \quad (17)$$

Where $\overline{X^*}.X$ and $X^*.\bar{X}$ represent respectively a rising and a falling edge of an order *X*.

$$CSs_3 = C^*_{filter}.\overline{C_{filter}}.\overline{fsc_{filter}} = 0 \quad (13)$$
$$CSs_4 = O^*_{filter}.\overline{O_{filter}}.\overline{fso_{filter}} = 0 \quad (14)$$

3.3. Results and Key Performance Indicators

A first analysis shows that the system is detectable in a bounded delay with certainty for the defined fault partitions. Indeed, all labels are represented in an abnormal state. However, the system is non-diagnosable with certainty. 10 labels on 13 possible are isolated with certainty (one unique label), 3 labels are with an ambiguity. For example, it is not possible to isolate with certainty states with labels {F1, F13} and {F3, F13}. Diagnostic Coverage (DC) is the ratio of the probability of detected dangerous failures (dd) to the probability all the dangerous failures (d). This meaning of the term DC is common to (ISO13849-1) and (IEC/EN 62061). For the valve, the DC is to 76.9%. The standard ISO13849-1 divides DC into four basic ranges: i) <60% = none, ii) 60% to <90% = low, iii) 90% to <99% = medium and iv) 99%+ = high. Consequently, another rule must be present to improve it and to guarantee complete diagnosability notion as defined in (Lin, 1994).

Table 1 presents a comparison between solutions with or without filter and/or diagnosers by simulation of the 13 faulty events under ProcesSim (http://processim.hecfh.be/). Thirteen scenarii have been exploited to obtain these results. With no filter, the valve system is under blocked behavior in 8 cases, into a degraded mode in 1 case and induces a defect situation for 4 cases. We can see that the tolerant situation disappear with the use of the filter only because its purpose is to ensure a safety behavior. When the FTC solution is used, the degraded mode is tolerant to 4 faulty events and above all, it decreases 2 cases of defect situations.

The proposed FTC approach has not been extended on all CISPI platforms yet. But a study has been done on a sub-system composed of 2 automatic valves, one pump and 2 tanks. Another point of view can be also to evaluate the steady state transition probabilities as a KPI. Indeed, a repetitive sequence of normal events can provide an indicator of the system behavior. For the moment, this remark is not treated in these works.

Table 1: Comparison with and without FTC solution

	Diag No Filter	No Diag Filter	FTC (Diag and Filter)
Blocked	8	9	7
Tolerant	1	0	4
Defect	4	4	2

4. CONCLUSION

A Fault Tolerant Control approach is presented around an interaction between diagnosers and filtering control. Diagnosis design is refined using enriched information from the real implemented control rules (control + violated constraints of the filter) while control filter benefits from using diagnose information to adapt its set of constraints according to reliable raw or constructed information.

In future works, when diagnosers detect a fault on a component or when the filter detects a mistake on the controller, a significant explanation must be given to a human operator to choose the best policy. A graduated explanation with potential consequences is to return. As last remark, the control filter has been implemented and extended to control design pattern on a real complex system called *CellFlex* at the University of Reims (www.univ-reims.fr/meserp/).

REFERENCES

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2003). Diagnosis and Fault-Tolerant Control. *Springer-Verlag*.

Brown D.W. and Vachtsevanos G.J. (2011). A Prognostic Health Management Based Framework for Fault-Tolerant Control. *Annual Conference of the Prognostics and Health Management Society* (PHM'11), Montreal, Quebec, Canada.

Cassandras C.G., Lafortune S. (1999). Introduction to discrete event systems. *Kluwer Academic Publishers*, Dordrecht.

Debouk R., Lafortune S. et Teneketzis D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete events systems. *In Journal of Discrete Event Dynamical System : Theory and Application*. pp.33-86.

Faure J-M., Lesage J-J. (2001). Methods for safe control systems design and implementations. *10th IFAC*

Symposium on Information Control Problems in Manufacturing, INCOM'2001, Vienna, Austria.

IEC/EN 62061. Safety of machinery: Functional safety of electrical, electronic and programmable electronic control systems. (2005).

ISO13849-1. Safety of machinery. Safety-related parts of control systems. General principles for design. (2006).

Kan John P., Grastien A. and Pencolé Y. (2010). Synthesis of a Distributed and Accurate Diagnoser. *21st International Workshop on the Principles of Diagnosis* (DX'10), Portland, Oregon, USA.

Lin F. (1994). Diagnosability of Discrete Event Systems and its Applications. *In Discrete Event Dynamic Systems*, 4, *Kluwer Academic Publishers*, Boston, USA.

Marangé P. (2008). Synthèse et filtrage robuste de la commande pour des systèmes manufacturiers surs de fonctionnement. PhD of the University of Reims Champagne-Ardenne.

Nke Y., Lunze J. (2011). Online control reconfiguration for a faulty manufacturing process. *3rd International Workshop on Dependable Control of Discrete Systems* (DCDS'11), Saarbrücken, Germany.

Paoli A., Sartini M. and Lafortune S. (2011). Active fault tolerant control of discrete event systems using online diagnostics. *Automatica*, Vol. 47, pp.639-649.

Philippot A. and Carré-Ménétrier. V. (2011). Methodology to obtain local discrete diagnosers. *3rd International Workshop on Dependable Control of Discrete Systems* (DCDS'11), Saarbrücken, Germany.

Qiu W. (2005). Decentralized/distributed failure diagnosis and supervisory control of discrete event systems, PhD of the Iowa State University, USA.

Ramadge G., Wonham W. M. (1989). The control of discrete event systems, *Proc. IEEE, Special issue on DEDSs*, 77, pp.81-98.

Riera B., Annebicque D., Gellot F., Philippot A., Benlorhfar R. (2012). Control synthesis based on logical constraints for safe manufacturing systems. *14th IFAC Symposium on INformation COntrol problems in Manufacturing* (INCOM 2012), Bucarest, Romania.

Riera B., Coupat R., Philippot A., Gellot F. and Annebicque D. (2014). Control design pattern based on safety Boolean guards for manufacturing systems: application to a palletizer. *12th IFAC-IEEE International Workshop On Discrete Event Systems* (WODES'14), France.

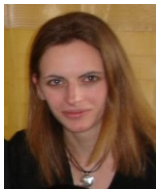
Roussel J.M. and Faure J.M. (2002). An algebraic approach for PLC programs verification. *In Proceedings of 6th international Workshop On Discrete Event Systems*, Zaragoza, Spain, pp.303–308.

Sampath M. (1995). A Discrete Event Systems Approach to Failure Diagnosis. PhD of the University of Michigan, Michigan, USA.

Wang, Y., Yoo, T. S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems*, Vol.17(2), pp233-263.

BIOGRAPHIES

Philippot Alexandre was born in France in 1979. After a Master degree in Systems Optimization and Safety, he achieved a PhD in Diagnosis of Discrete-events Systems at the University of Reims Champagne-Ardenne in France in 2006. Currently, he is associated professor at the University of Reims Champagne-Ardenne (URCA) and realises its research at CReSTIC Laboratory (research center in Sciences and Technologies on Information and Communication). Research interests: Discrete Event Systems, Fault diagnosis, Modeling, Supervisory Control Theory, Optimal Control.



Pascale Marangé was born in 1982 in France. After a Master degree at the University of Reims Champagne-Ardenne (URCA) in France, she achieved a PhD in Control synthesis of Discrete-events Systems at URCA in 2008. Currently, she is associated professor at Lorraine University and realizes its research at Nancy Research Center for Automatic Control (CRAN). Her research interests include Verification and Validation, Dependability of Discrete-Event Systems, Control synthesis.

François Gellot is associated professor at the University of Reims Champagne-Ardenne (URCA) France. He was born in France in 1965. He achieved a PhD in analyze and simulation of Petri nets (1994). Its research interests include supervisory control, discrete events systems modeling and Verification and Validation of PLC program.

Jean-François Pétin is a Professor at Lorraine University and realizes its research at Nancy Research Center for Automatic Control (CRAN). Its research interest concerns Dependability of Discrete-Event Systems, Control synthesis, Verification & Validation.

Bernard Riera received the Ph.D. degree in Automatic Control from the University of Valenciennes (UVHC), in 1993. He is a Professor of Control Engineering at the University of Reims Champagne-Ardenne (URCA) France, a Researcher at the CReSTIC (research center in Sciences and Technologies on Information and Communication) and associate director of the CReSTIC. Its research interests include supervisory control, supervisory support systems, discrete events and hybrid systems modeling.

A Novel Feature Extraction Method for Monitoring (Vehicular) Fuel Storage System Leaks

Fling Tseng[1], Imad H. Makki[2], Dimitar P. Filev[3], and Ratna Babu Chinnam[4]

[1,2,3]*Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48121, USA*

ftseng@ford.com
imakki@ford.com
dfilev@ford.com

[4]*Department of Industrial and Systems Engineering, Wayne State University, Detroit, MI 48202, USA*

ratna.chinnam@wayne.edu

ABSTRACT

System state determination with incomplete sensory information set proved to be a technically challenging problem. In this paper, authors tackle a problem of this type associated with vehicle fuel storage systems and proposed a novel feature extraction method. Federal and state regulations require fuel storage leak detection mechanism to be conducted periodically and regulate its execution rate and performance to ensure effective emission controls. Being able to robustly determine a fuel storage system's state in terms of its effectiveness of fuel containment is therefore of great importance to all vehicle original equipment manufacturers (OEM). Prevailing practice in the industry utilizes a method relevant to natural vacuum phenomenon and is loosely associated with ideal gas law. Commonly referred to as "Entry Conditions" in in-vehicle monitoring design literature, major noise factors go through stringent pre-monitoring evaluations before monitoring program execution to ensure ideal test conditions. Differences in ambient conditions compounded with varying customer drive cycle patterns present great challenge to existing monitor designs for the purpose of leak detection. In addition, prevailing practices of evaluation in-tank fuel pressure and temperature information are generally conducted with surrogate or estmiated temperature information due to the absence of in-tank temperature sensor. All this calls for an alternative feature calculation and detection method that are less sensitive to known noise factors, can operate with incomplete sensory information yet being able provide similar or improved detection capability. In this paper, we put the main focus on the derivation of a novel method of feature calculation for the purpose of detecting presence of a leak in a fuel storage tank.

Fling Tseng et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

1. INTRODUCTION

Murvay (Murvay, 2012) studied state-of-the-art development in terms of hardware (including pressure, acoustic, remote and reflective sensing) and software methods for gas leak detections. It was concluded that a hybrid approach to take advantage of cost effective hardware setup (high localization accuracy) with fast improving software methods (real-time detection capability) would be highly recommended. It also suggests that investment in a hybrid approach may be more cost effective in the long term as software capability enhancements may offset the effect of aging hardware, reducing the need for a complete revamp of leak detection setup, something very cost prohibitive. Zhou (Zhou, 2011) proposed a Bayesian Belief Rule Based (BRB) system where subject expert knowledge and real-time information are incorporated to incrementally improve the performance of the system. Such a combination of human knowledge and data driven refinement to the model is suitable to deal with ever increasingly complex real-world problems. Ghazali's work (Ghazali, 2012) focused on instantaneous frequency analysis (IFA), where comparisons between Hilbert transform (HT), Normalized HT (NHT), Direct Quadrature (DQ), Teager Energy Operator (TEO) and Cepstrum performed on pressure transients (opening a valve or stopping a pump) within a live distribution network were conducted. A detection method that includes multiple modeling techniques was proposed by (Mandal, 2012). They apply rough set theory and artificial bee colony (ABC) trained SVM (Support Vector Machine) to carry out classification tasks in two stages and yielded robust performance when compared with PSO (particle swarm optimization) and

EPSO (enhanced particle swarm optimization) based learning methods.

Leak detection mechanism as part of an overall emission control strategy is gaining importance in recent years. As countries are increasingly pledging reduced carbon footprints, one of the main focuses was to incrementally reduce and eventually eliminate allowable fuel vapors escaped to the ambient air. In the United States, ongoing efforts from Environmental Protection Agency (EPA) and California Air Resources Board (CARB) requires consumer vehicle original equipment manufacturers (OEMs) to equip their products with leak detection monitors to improve monitoring capabilities within a given timeframe (State of California Air Resources Board, 2012). In the meantime, on the field performances are under federal and state level regulations subject to audits. If sampled results are deemed unsatisfactory, fines or even voluntary recalls could be imposed. These penalties are undesirable as they undermine an OEM not only financially but could also negatively affect brand image that take years to even decades to recover if such incidents occur.

Emmission related monitors generally reside in the powertrain control module (PCM) therefore contraints such as A. During calculation memory requirement, B. Computational efficiency and C. Compactness of the code often need to be carefully evaluated due to implications in terms of cost and practicality during implementation phase. In this paper, authors focus on describing a fundamentally different way of extracting information from the in-tank pressure signal stream as it is one of most critical parts of an overall redesign of an in-vehicle monitor. More specifically, we will cover a recursive approach to enable monitor design engineers to have access to physically meaningful probability density function (PDF) type of information continuously in the form of a recursively updated histogram or discretized probability density function (DPDF) from normalization performed on an obtained discretized relative frequency function (DRFF). Feature calculations are performed from evaluation of certain specific bin(s) of the DPDF from which decisions can be made about the fuel tank's status with repect to the presence of a leak. Technique descibed in (Syed, 2009) utilizes a low pass filter (LPF) implementation to extract driver (non-conditional / overall) behavioral information for adaptation of an in-vehicle advisory system. When applied to scenarios where possible alternatives do exist, such calculation produces conditional relative frequency (RF) information which is a precursor of probabilistic information. In (Filev, 2011), organization and conditional updates of trip specific RF values enable the creation of a context senstive predictive system. Proposed feature exraction method strictly operates in the probabilistic space. It represents a significant step forward and a crucial enabling element to improve from prevailing pactice of evaluation of pressure signal (or its manipulated version) alone (Wong, 2003 and Jentz, 2013). Our preliminary analysis suggests proposed feature calculation produces meaningful and promising results. The investigation of promising alternative feature calculations as the one described in this paper is an important first step that shall shed more light on how to redesign a leak detection monitor in the future.

The rest of the paper is organized as the following. In section 2, current prevailing practices in the industry will be discussed where most OEM's approach can be understood as solving a classification problem (leak vs no leak) with a single feature commonly derived from in-tank pressure signal. In section 3, the derivation and computation procedure of obtaining a continuous measure of the content of in-tank pressure signal stream in the form of DPDF. In addition, proposed feature calculation from DPDF vector is desribed in detail. Section 4 covers a simple threshold determination based classification process utilizing the feature calculation described in Section 3 and preliminary results are presented. We conconlude current findings and future work in section 5 followed by cited references.

2. INDUSTRY PRACTICE FOR VEHICULAR LEAK DETECTION

Prevailing principle of fuel storage leak detection design relies on well-known "Ideal Gas Equation", which states the governing relationship between system pressure and temperature given certain characterizing constants or a lumped product is known or estimated (Wong, 2003 and Jentz, 2013). Determination of the presence of a leak in the fuel storage system is carried out by evaluation of whether expected pressure change is met within certain threshold (2005, McLain). Due to its evaporative nature, gasoline vapor / liquid state transition activities does not warrant the direct use of the ideal gas equation, therefore, monotor specific "Entry Condition" evaluations have to be carried out before monitoring program execution.

After vehicle key-off, when entry conditions are met, the system is then sealed by operation of certain actuators such as valves. In this phase, in-tank pressure signal is kept alive for evaluation against thresholds that are dynamically adjusted to ambient as well as preceding driving conditions that led to the current stop. During all this time, parallel evaluations of certain run time parameters are common to reduce false state determinations and total engine-off battery draw. When it is deemed an effective determination cannot be reached, execution could self-abort without making a determination as to the system's state. A set of built-in counters are required by law to be in place to keep track of how often a monitor runs against scenarios it is required to do so. The ratio of leak / no leak versus total number of successfully full executions are also being tracked. These values are subjected to insepctions of government agencies and OEM's periodically.

Abovementioned leak detection process can be understood as carrying out a classification procedure with a main feature that is commonly derived from pressure sensor information. The goal of these leak detection monitors is to produce a leak indicator value [0, 1] in which 0 represents no leak state and 1 represents presence of a sizable leak. The original pressure value is subjected to futher common signal procesing methods such as signal smoothing, clipping and flipping. Other common modifications may also include multiple scalers associated with ambient / vehicle conditions. After a series of manipulations, comparison is performed with thresholds resulted from calibrations conducted with a sweep of main noise factors spaces. Different from abovementioned commonly used feature, section 3 describes in detail a recursive procedure continously measure in-tank pressure content in the form of DPDF from which feature(s) will be calculated for the purpose of leak detection.

3. FEATURE DERIVATION FROM PROBABILITY DENSITY CURVE FOR CLASSIFICATION PURPOSE

The first step in solving a classification problem generally has to do with identification of effective features. Feature extraction serves at least following purposes: 1) Obtaining informative representation of data, 2) Dimensionality reduction, and 3) Reduction in noise and redundancy. Common feature extraction methods can be grouped into the following categories: 1) Time series based features, 2) Statistics based features, 3) Frequency based features, 4) Mixed domain features, and 5) Model based features. For some applications (e.g., vibration analysis), expert and domain knowledge play important roles in guiding the methodology and techniques involved in the feature extraction process. While certain calculation and data transformation may be common (e.g., Fourier Transform for accelerometer sensing signals), such practice may produce signatures associated with certain frequency range. Depending on subject problem of interest, simple data smoothing, deterministic or moving data window scheme or windowed data overlay techniques may be imposed as part of a feature extraction procedure. Details regarding signal and feature selection process are out of the scope of this paper.

Different from common practice, the authors performed data analysis focused on signatures revealed from the probability density function of in-tank pressure changes. This is one of the signals typically kept "alive" during leak detection monitoring phase after the engine has been turned off and the system has been sealed. More specifically, we developed a non-parametric method to continuously extract signatures indicative of the existence of a leak in a presumably sealed setting. The rationale is that change in overall pressure is a consequence of accumulated pressure (rate) changes. We apply procedures to obtain dprobability distribution function in a discretized form from the frequentist's point of view (of relative frequency). This is procedure is implemneted with a low pass filter (LPF or 1st order exponential smoothing). After initialization phase (where a number of initial signal samples have been observed), proposed method gives a continuous output of the DPDF with predefined partitions. Resolution a DPDF is dependent on pre-determined signal range and number of partitions within that range.

Conceptually, proposed implementation is identical to the creation of a histogram with a moving data windown given some continuously incoming data stream; the counting procedure is carried out by a LPF in which its learning rate controls the size of the moving data window. The crisp partitions within specified signal range act as "competing and possible" scenarios or alternatives where we impose a "winner takes all" rule for relative frequency (RF) updates for all partitions involved. Through this updating rule, the increment of the relative frequency occurs only for one partition at a time while the rest of the competing partitions receive negative updates. At any given time, a DPDF is obtained by normalizing most recent DRFF with the summation of its elements. Details regarding this process are described next.

3.1. Recursive Estimation of Discretized Relative Frequency Function (DRFF) as Predecessor of Discretized Probability Density Function (DPDF)

3.1. Recursive Estimation of Discretized Relative Frequency Function (DRFF) as Predecessor of Discretized Probability Density Function (DPDF)

From a frequentist's point of view of probability, probability density function (PDF) comes from obtaining a histogram-like vector (of very fine granulaity or partition), namely a DRFF. After a normalization procedure, a DPDF is obtained and the summation of its content should be 1 (sum of total probability of 1). In the simplest case, the first step in obtaining DRFF vetor is to partition a signal's value space into smaller non-overlapping ones. For example, if a signal X takes values from 0 to 10, an example of such a partition would be to define 10 partitions of the signal space that spans the following consecutive intervals or bins: $0 \le x < 1$, $1 \le x < 2$, $2 \le x < 3$ … $9 \le x \le 10$. As a result, they represent mutually exclusive scenarios or value range alternatives regarding numeric content of signal X at any given moment. When a specific component of data stream of signal x is being evaluated, only one of the the alternatives will receive the increment in count from the fact current x's value falls into a corresponding region while other alternatives will receive negative updates. From (Syed, 2009), the construction of a count based histogram can be approximated recursively with an exponentially weighted moving average (EWMA) formulation where counts are replaced with relative frequencies (RF). When such implementation is in place, content captured in an interval in DRFF represents a relative frequency value corresponds to

the total number of occurances relative its alternatives (other intervals). For example, if α is 0.05 the moving window is approximately 1/0.05 = 20 meaning that at any given moment the DRFF preserves information from the most recent past 20 observations of signal X. The process of obtaining DRFF can be represented by following equation:

$$DRFF_i(t) = (1-\alpha)\cdot DRFF_i(t-1) + \alpha \cdot Flag_i(t) \quad (1)$$

where ⟦ DRFF⟧ _i denotes relative frequency of a partition enclosed by its lower and upper limits, α denotes the learning (0≤α≤1), and ⟦ Flag⟧ _i denotes a binary flag value of 0 or 1 indicating whether current value of X falls into the regrion defined by the i'th region. All partitions of DRFF go through exactly one update during the evaluation of one incoming signal value with Eq (1) and all but one of the partitions will experience a value increment due to the use of "winner takes all" updating rule.

DPDF is obtained by normalization procedure performed on DRFF with following equation:

$$DPDF_i(t) = {DRFF_i(t)}\Big/{\sum_{i=1}^{N} DRFF_i(t)} \quad (2)$$

With equation (2), DPDF is obtained from updated DRFF from which subsequent feature calculation will be performed.

A numerical example comparing LPF vs actual counts based DPDF is shown in the Figure 1.

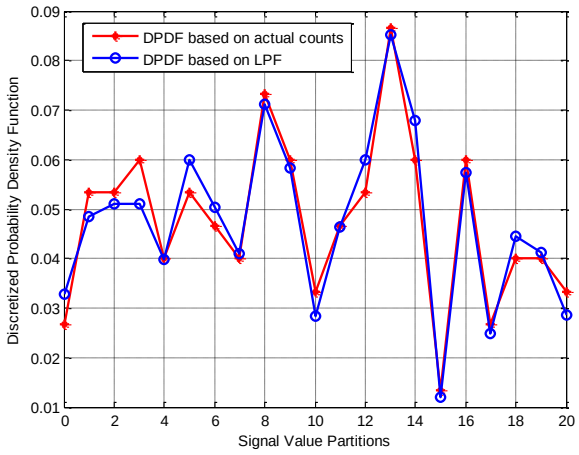


Figure 1: Comparison of recursively obtained DPDF vs Actual Count generated DPDF

In Figure 1, a total of 150 random integers ranging from 0 to 20 were populated.

3.2. Extracting Probability Density Content from In-Tank Pressure

3.2.1. Focus of 1st Sealed Stage

During experiments to generate representative datasets, the fuel storage system (fuel tank) goes through a series of state transitions that either expose or seal the system from the atmosphere. The rationale for the transitions contains proprietary information, and hence, will not be discussed here. Our research development focused on the 1st seal stage of all datasets. The reason being that subsequent changes are dependent on information collected during a prior state, making comparison between datasets not realistic. In addition, we identified that the early stage in the 1st sealed phase is much more informative; therefore, we will focus on data collected in the first 300 seconds of each dataset. In addition, we have found that the contrast (separation) between classes reduced for the proposed method very quickly after 300 seconds into the 1st sealed phase.

3.2.2. Pressure Change between Samples vs Pressure Change Rate

The determination that a system has entered its 1st sealed state is conducted by monitoring a set of flags associated with actuators' (valves) states that could be either open or closed. When the system is deemed to have entered its 1st sealed phase, the difference between previous and current in-tank pressures (inch mercury) is calculated continuously. Since our data collection system collects information at a (almost) constant rate of 10 Hz (every 100 milliseconds), pressure change rate in this case is proportional to pressure change between samples, and therefore, we omit the normalization division operation to simplify the calculation.

3.2.3. Obtaining Vector Probability Density Content

First of all, the signal numeric space is defined as 100 equally spaced (0.0003) partitions ranging from -0.015 to 0.015. α is set to be 1/500 or 0.002, which is equivalent of imposing a moving data window containing the last 500 samples as it moves through the data stream. Since the normalization process effectively only scales DRFF through division of its element sum, the overall shape DRFF will be identical to DPDF. A snapshot of DPDF serves as a visual example is shown is Figure 2 according to partitions based on aforementioned definition.

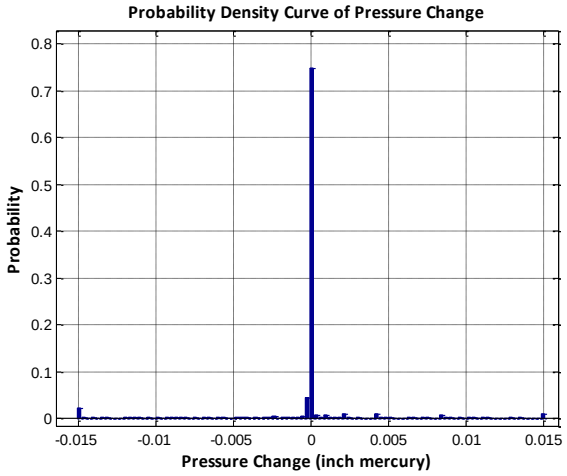


Figure 2: DPDF obtained from normilzation of DRFF covering value range [-0.015, 0.015]. Each partition is of the width of 0.0003.

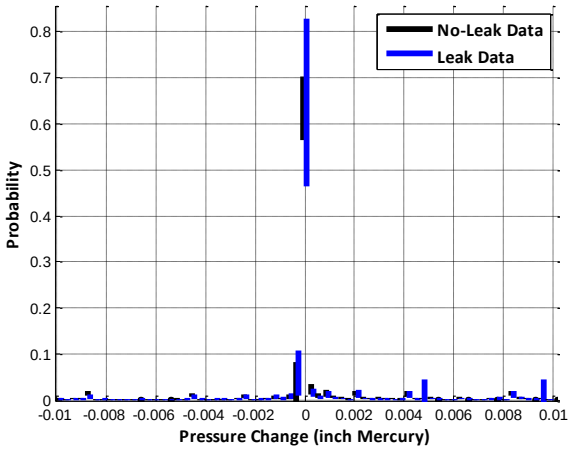


Figure 3: Visualation of DPDF content of Leak (Blue) vs No-Leak (Black) Datasets. For each partition, upper bound / lower bound are obtained with μ+3σ and μ-3σ to visualize the location of the mean value and its spread simultaneously.

3.2.4. Identification of Effective Features from DPDF for Classification Purpose

From Figure 2, we noticed an interesting fact that close to 75% of pressure change readings are assigned to the partition centered at 0 for this particular experimental dataset. This is not a coincidence but a result of the sensitivity of the pressure sensor in the existing product.

The next step is to perform the same computational procedures to all datasets. With predefined partitions as described in 3.2.3, resuling DPDF from all datasets are inherently of the same size making it straightforward for us to calculatae the mean and standard deviations separately for two populations: leak vs no leak datasets. As a result, we obtained two sets of means and standard deviations for each partition using following equations:

$$\overline{\mu}_{DPDF_i} = \frac{\sum_{j=1}^{K} DPDF_{i,j}}{K} \quad (3)$$

$$\overline{\sigma}_{DPDF_i} = \sqrt{\frac{\sum_{j=1}^{K} (DPDF_{i,j} - \overline{\mu}_{DPDF_i})^2}{K-1}} \quad (4)$$

i denotes a particular partition, j denotes a dataset and K represents total number of datasets. Since we peforms such calculations for leak and no leak datasets separately, K will take different values if we have an unbalanced datasets where total numbers of leak and no leak datasets are different. From (3) and (4), we obtained population mean and standard deviation of each defined partition. We employ the well-known 6σ definition to show the range spans μ-3σ and μ+3σ for each partition separately for leak (blue line) vs no-leak (black line) datasets as shown in Figure 3.

Selective use of content from DPDF partitions for the purpose of distinguishing between leak and no leak (classification) datasets need to fulfill at least following criteria: 1) Potential content from a partition should exhibit class separation potential and 2) Potential content from a partition should have likelyhood of taking values (non-zero). The first criteria suggests that patterns shown in DPDF should have some class separating capability such as μ_leak ≤ μ_(no-leak) such as the partition around 0.015 as shown in Figure 4. Or, as shown in Figure 3, the partition around zero that the spreads are different between classes, which indicates standard deviations of no-leak datasets may be generally smaller than those of leak datasets. The second criteria has to do with selection of content elements that will take value in the sealed process making sure such content will available to determine the overall system's state in terms of the presence of a leak. This criteria is a basic yet a necessary one to ensure content availability of a partition from DPDF from which subsequent feature calculations are based on.

Following aforementioned criteria, we will mainly focus on the features extracted from DPDF partition near the zero. This is due to the overall low DPDF values of almost all other partitions indicating risks of them to take value on a consistent basis.

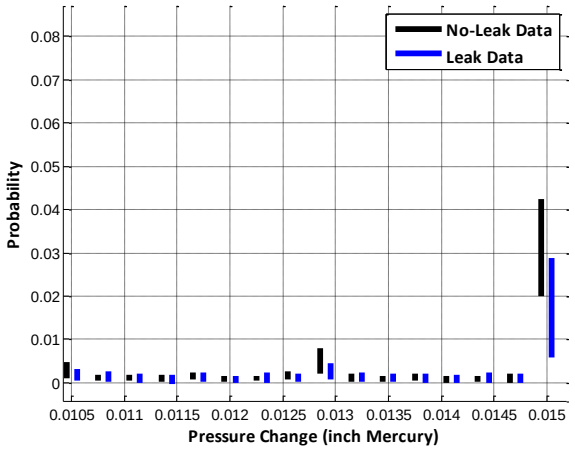


Figure 4: Zoom-in view of Figure 3 focus on partitions on the positive side. For partition centered at 0.015, with some overlapping the means of leak vs no leak populations exhbit certain level of difference.

3.2.5. Continuous Evaluation of DPDF Content Derived Features for Leak vs No-Leak System State Determination

One advantage of using recursive equation for feature extraction is the enablement of continuous assessment of the system of interest. In Figure 5, DPDF partition content around zero for multiple leak (upper figure) and no leak (lower feagure) datasets (as described in 3.2.4) are shown in time domain where we can visually validated the continuous class separation capability.

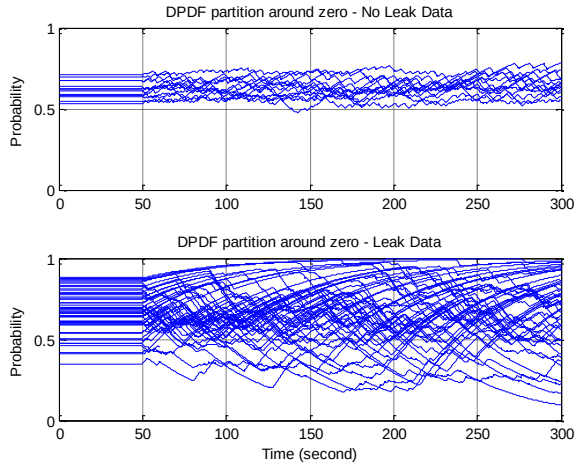


Figure 5: Continuous Evaluation of Content derived from DPDF partition around zero. DPDF content (Y-axis) as shown is presented in terms of probability where 1 equals 100%. Upper figure includes only datasets with no leak. Lower figure includes only datasets with leak.

4. CLASSIFICATION WITH A SIMPLE THRESOLD SETTING AND RESULTS

Existing datasets to test out the method contains data streams that are collected for calibration purpose of existing strategies. Due to current monitor's design, datasets collected for this purpose tend to put more focus on datasets with leaks. There are 14 data files labeled as system that has been verified to have no leak and 53 data files that have induced leak. When applied to existing monitor, nearly half of all dataset will be thrown out without being evaluated due to failures to pass one of the entry condtions in place.

For simplication purpose, we will refer to $DPDF_0$ for the probability value obtained from the partition around zero. We employ method described above to calculate $DPDF_0$ continuously at a particular common execution phase of current strategy where the system was commanded to be sealed.

$$\mu_{MAX\ of\ DPDF_0} = \frac{\sum_1^k \max(DPDF_{0,k})}{k} \quad (5)$$

$$\mu_{min\ of\ DPDF_0} = \frac{\sum_1^k \min(DPDF_{0,k})}{k} \quad (6)$$

$$\sigma_{MAX\ of\ DPDF_0} = \sqrt{\left(\frac{\sum_1^k (\max(DPDF_{0,k}) - \mu_{MAX\ of\ DPDF_0})^2}{k-1}\right)} \quad (7)$$

$$\sigma_{min\ of\ DPDF_0} = \sqrt{\left(\frac{\sum_1^k (\min(DPDF_{0,k}) - \mu_{min\ of\ DPDF_0})^2}{k-1}\right)} \quad (8)$$

The characterization of PDC0 from no leak dataset involves using 10 no leak data files. From these files, means and standard deviations of maximum and minimum values of each PDC0 profiles are obtained. Currently, upper and lower thresholds are estimated separately taking the common form as the following:

$$Threshold_{Upper} = \mu_{MAX\ of\ DPDF_0} + k_1 \cdot \sigma_{MAX\ of\ DPDF_0} \quad (9)$$

$$Threshold_{Lower} = \mu_{min\ of\ DPDF_0} + k_2 \cdot \sigma_{min\ of\ DPDF_0} \quad (10)$$

For each dataset, $DPDF_0$ profiles are evaluated continuously against Threshold_Upperand Threshold_Lower. System is deemed to be leaky if at any given time "either" threshold is exceeded.

Identification of thresholds k1 and k2 are performed with following procedure. We divide both datasets with leak and datasets with no leak into 2 equal sized groups (training and validation). As a result, each group contains 7 no leak datasets. In addition, training group contains 26 leak datasets and validation group contains 27. We enumerate k1 and k2 values between -3 to 3 with 0.1 increments to identify potential pairs of k1 and k2 producing reasonable results. In this case, we define a reasonable performance as being able to at least classify all no leak datasets correctly. After that, passing pairs are ranked based on their detection

rate for leak datasets. In this process we found that among 31*31 = 961 pairs there exist 20 pairs of k1 and k2 to have the same results. For these pairs, the overall prediction rates are the same at 100% meaning all leak and no leak datasets were identified correctly. They tend to have k1 around 0.9 ~ 0.18 and k2 to be either -0.7 or -0.8.

Table 1. k1 and k2 pair test sequence and detection rates for leak datasets, no leak datasets and when combined.

Testing Sequence	K1	K2	Detection Rate (%) - No Leak	Detection Rate (%) - Leak	Detection Rate (%) - all
1434	0.9	-0.8	100.0%	100.0%	100.0%
1435	1	-0.8	100.0%	100.0%	100.0%
1436	1.1	-0.8	100.0%	100.0%	100.0%
1437	1.2	-0.8	100.0%	100.0%	100.0%
1438	1.3	-0.8	100.0%	100.0%	100.0%
1439	1.4	-0.8	100.0%	100.0%	100.0%
1440	1.5	-0.8	100.0%	100.0%	100.0%
1441	1.6	-0.8	100.0%	100.0%	100.0%
1442	1.7	-0.8	100.0%	100.0%	100.0%
1443	1.8	-0.8	100.0%	100.0%	100.0%
1495	0.9	-0.7	100.0%	100.0%	100.0%
1496	1	-0.7	100.0%	100.0%	100.0%
1497	1.1	-0.7	100.0%	100.0%	100.0%
1498	1.2	-0.7	100.0%	100.0%	100.0%
1499	1.3	-0.7	100.0%	100.0%	100.0%
1500	1.4	-0.7	100.0%	100.0%	100.0%
1501	1.5	-0.7	100.0%	100.0%	100.0%
1502	1.6	-0.7	100.0%	100.0%	100.0%
1503	1.7	-0.7	100.0%	100.0%	100.0%
1504	1.8	-0.7	100.0%	100.0%	100.0%

Using these pairs we obtained best overall detection rate of 88% that is slightly worse yet very similar to the result of the original leak monitor. The two k1 and k2 pairs produced best result during validation have the same k1 to be 0.9 and k2 to be -0.7 and -0.8 respectively at sequence #1434 and #1495. One thing to note is that application of the proposed method does not require a large set of entry conditions before monitoring procedures being executed. In other words, proposed feature calculation with a simple thresholding method result in significantly improved monitor applicability in comparison with current design.

Table 2. k1 and k2 pair validate sequence and detection rates for leak datasets, no leak datasets and when both are combined.

Validation Sequence	K1	K2	Detection Rate (%) - No Leak	Detection Rate (%) - Leak	Detection Rate (%) - all
1434	0.9	-0.8	85.7%	88.9%	88.2%
1435	1	-0.8	85.7%	85.2%	85.3%
1436	1.1	-0.8	85.7%	85.2%	85.3%
1437	1.2	-0.8	85.7%	81.5%	82.4%
1438	1.3	-0.8	85.7%	81.5%	82.4%
1439	1.4	-0.8	85.7%	81.5%	82.4%
1440	1.5	-0.8	85.7%	77.8%	79.4%
1441	1.6	-0.8	85.7%	77.8%	79.4%
1442	1.7	-0.8	85.7%	77.8%	79.4%
1443	1.8	-0.8	85.7%	77.8%	79.4%
1495	0.9	-0.7	85.7%	88.9%	88.2%
1496	1	-0.7	85.7%	85.2%	85.3%
1497	1.1	-0.7	85.7%	85.2%	85.3%
1498	1.2	-0.7	85.7%	81.5%	82.4%
1499	1.3	-0.7	85.7%	81.5%	82.4%
1500	1.4	-0.7	85.7%	81.5%	82.4%
1501	1.5	-0.7	85.7%	77.8%	79.4%
1502	1.6	-0.7	85.7%	77.8%	79.4%
1503	1.7	-0.7	85.7%	77.8%	79.4%
1504	1.8	-0.7	85.7%	77.8%	79.4%

5. CONCLUSION AND FUTURE WORK

We have proposed a novel method to obtain an effective feature from discretized probabilistic density function continuously. Using a simple threshold mechanism, different thresholds are setup such that exceeding either one indicates the presence of a leak in the system. Compared with existing strategies that use a set of entry conditions to determine whether to execute a test or not, proposed method produced similar detection rate while significantly increases applicability (no entry conditions has to be imposed).

In addition to the simple threshold setting approach presented in this paper, continuing effort will be focused on evaluating the usage of more effective data classification methods such as SVM, Bayesian Classifiers, Fuzzy Classifiers or LVQ with proposed feature. The eventual goal is to redesign computation procedures that minimizes false positives/negatives (robustness), enhances system performance (performance) in real-world settings with broad coverage (applicability). We believe continual effort in this field will ensure future technical advancement in this fundamental yet critical aspect in emission reduction and control.

REFERENCES

Murvay, Pal-Stefan, and Ioan Silea. (2012). *A survey on gas leak detection and localization techniques.* Journal of Loss Prevention in the Process Industries 25.6 pp.966-973.

Zhou, Z. J., Hu, C. H., Xu, D. L., Yang, J. B., & Zhou, D. H. (2011). *Bayesian reasoning approach based recursive algorithm for online updating belief rule based expert system of pipeline leak detection.* Expert Systems with Applications, 38(4), 3937-3943.

Ghazali, M. F., Beck, S. B. M., Shucksmith, J. D., Boxall, J. B., & Staszewski, W. J. (2012). *Comparative study of instantaneous frequency based methods for leak detection in pipeline networks.* Mechanical Systems and Signal Processing, 29, pp. 187-200.

Mandal, S. K., Chan, F. T., & Tiwari, M. K. (2012). *Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained SVM.* Expert Systems with Applications, 39(3), 3071-3080.

State of California Air Resources Board (2012). *California Evaporative Emission Standards and Test Procedures for 2001 and Subsequent Model Motor Vehicles.*

State of California Air Resources Board (2012). *California Refueling Emission Standards and Test Procedures for 2001 and Subsequent Model Motor Vehicles.*

Filev, D., Tseng, F., Kristinsson, J., & McGee, R. (2011). *Contextual on-board learning and prediction of vehicle destinations.* In Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on. pp. 87-91

Wong, Kevin C., and William Western. (2003) *Engine off natural vacuum leakage check for onboard diagnostics.* U.S. Patent No. 6,550,316. 22

Jentz, R. R., Clemens, D. A., Sipes, A. I., Dudar, A. M., & Kluzner, M. I. (2013). *System and method for performing evaporative leak diagnostics in a vehicle.* US 8725347 B2

McLain, Kurt D., and Gregory E. Labus. (2005). *Apparatus and method for fuel vapor leak detection.* US 6880383 B2

Syed, Fazal U., Dimitar Filev, Fling Tseng, and Hao Ying. (2009). *Adaptive real-time advisory system for fuel economy improvement in a hybrid electric vehicle.* In Fuzzy Information Processing Society, NAFIPS 2009. Annual Meeting of the North American, pp. 1-7.

Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset

Emmanuel Ramasso[1] and Abhinav Saxena[2]

[1] *FEMTO-ST Institute, Dep. AS2M/DMA, UMR CNRS 6174 - UFC / ENSMM / UTBM, 25000 Besançon, France*
emmanuel.ramasso@femto-st.fr

[2] *SGT Inc., NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA, 94035-1000, USA*
abhinav.saxena@nasa.gov

ABSTRACT

Benchmarking of prognostic algorithms has been challenging due to limited availability of common datasets suitable for prognostics. In an attempt to alleviate this problem, several benchmarking datasets have been collected by NASA's prognostic center of excellence and made available to the Prognostics and Health Management (PHM) community to allow evaluation and comparison of prognostics algorithms. Among those datasets are five C-MAPSS datasets that have been extremely popular due to their unique characteristics making them suitable for prognostics. The C-MAPSS datasets pose several challenges that have been tackled by different methods in the PHM literature. In particular, management of high variability due to sensor noise, effects of operating conditions, and presence of multiple simultaneous fault modes are some factors that have great impact on the generalization capabilities of prognostics algorithms. More than 70 publications have used the C-MAPSS datasets for developing data-driven prognostic algorithms. The C-MAPSS datasets are also shown to be well-suited for development of new machine learning and pattern recognition tools for several key preprocessing steps such as feature extraction and selection, failure mode assessment, operating conditions assessment, health status estimation, uncertainty management, and prognostics performance evaluation. This paper summarizes a comprehensive literature review of publications using C-MAPSS datasets and provides guidelines and references to further usage of these datasets in a manner that allows clear and consistent comparison between different approaches.

1. INTRODUCTION

In the past decade, the science of prognostics has fairly matured and the general understanding of health prediction problem and its applications has greatly improved. Both data-driven and physics based methods have been shown to possess unique advantages that are specific to application contexts. However, until very recently, a common bottleneck in development of data-driven methods was the lack of availability of run-to-failure data sets. In most real-world cases, data contain fault signatures for a growing fault at various severity levels but no or little data capture fault evolution all the way through failure. Procuring actual system fault progression data is typically time consuming and expensive. Fielded systems are, most of the time, not properly instrumented for collection of relevant data or are unable to distribute such data due to proprietary constraints. The lack of common data sets, which researchers can use to compare their approaches, has been an impediment to progress in the field of prognostics. To tackle this problem, a prognostics data repository was established (Saxena & Goebel, 2008). Several datasets have been since published that have been used by researchers around the world. Among these datasets are five datasets from a turbofan engine simulation model - C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) (Frederick, DeCastro, & Litt, 2007). By simulating a variety of operational conditions and injecting faults of varying degree of degradation, datasets were generated for prognostics development (Saxena, Goebel, Simon, & Eklund, 2008a). One of the first datasets was used for a prognostics data challenge at the PHM'08 conference. A subsequent set was then released later with varying degrees of complexity. These datasets have since been used very widely in publications for benchmarking prognostics algorithms.

The turbofan degradation datasets have received over seven thousand unique downloads in the last five years but algorithms developed using these have been published in only about seventy publications. Furthermore, in many publications it is not clear how authors are computing results and comparing with others. There has been a confusion and inconsistency in how these datasets have been interpreted and

Emmanuel Ramasso et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

used in many cases. Consequently, not all comparisons of performance can be considered valid. Therefore, this paper intends to analyze various approaches that researchers have taken to implement prognostics using these turbofan datasets. Some unique characteristics of these datasets are also identified that led to use of certain methods more often than others. Specifically, various differences among these datasets are pointed out. A commentary is provided on how these approaches fared compared to the winners of the data challenge. Furthermore, this paper also attempts to clear several issues so that researchers, in the future, can take these factors into account in comparing their approaches with the benchmarks.

The paper is organised as follows. In Section 2, the C-MAPSS datasets are presented. Section 3 is dedicated to the literature review. Section 4 presents a taxonomy of prognostics approaches for C-MAPSS datasets. Finally, Section 5 provides some guidelines to give a hand to future users in developing new prognostic algorithms applied to these datasets and in facilitating algorithms benchmarking.

2. C-MAPSS DATASETS

C-MAPSS is a tool, coded in the MATLAB-Simulink ® environment for simulating engine model of the 90,000 lb thrust class (Frederick et al., 2007). Using a number of editable input parameters it is possible to specify operational profile, closed-loop controllers, environmental conditions (various altitudes and temperatures), etc. Additionally, there are provisions to modify some efficiency parameters to simulate various degradations in different sections of the engine system.

2.1. Datasets characteristics

Using this simulation environment, five datasets were generated. By creating a custom code wrapper, as described in (Saxena, Goebel, et al., 2008a), selected fault injection parameters were varied to simulate continuous degradation trends. Data from various parts of the system were collected to record effects of degradations on sensor measurements and provide time series exhibiting degradation behaviors in multiple units. These datasets possess unique characteristics that make them very useful and suitable for developing prognostic algorithms.

1. Data represent a multi-dimensional response from a complex non-linear system from a high fidelity simulation that very closely models a real system.
2. These simulations incorporated high levels of noise introduced at various stages to accommodate the nature of variability generally encountered.
3. The effects of faults are masked due to operational conditions, which is yet another common trait of most operational systems.
4. Data from plenty of units is provided to allow algorithms to extract trends and build associations for learning system behavior useful for predicting RUL.

These datasets were geared towards data-driven approaches where very little or no system information was made available to PHM developers.

As described in detail in Section 3, the analysis on the publications using these datasets shows that many researchers have tried to make comparisons between results obtained from these similar yet different datasets. This section briefly describes and distinguishes the five datasets and explains why it may or may not be appropriate to make such comparisons. Table 1 summarizes the five datasets. The fundamental difference between these datasets is attributed to the number of simultaneous fault modes and the operational conditions simulated in these experiments. Datasets $\#1$ through $\#4$ incorporate an increasing level of complexity and may be used to incrementally learn the effects of faults and operational conditions. Furthermore, what sets these four datasets apart from the challenge datasets is the availability of ground truth to measure performance. Datasets $1-4$ consist of a *training set* that users can use to train their algorithms and a *test set* to test the algorithms. The ground truth RUL values for the test set are also given to assess prediction errors and compute any metrics for comparison purposes. Results between these datasets may not always be comparable as these data simulate different levels of complexity, unless a universal generalized model is available that regards datasets $1-3$ as special cases of dataset $\#4$.

The PHM challenge datasets are designed in a slightly different way and divided into three parts. Dataset $\#5T$ contains a *train set* and *test set* just like for datasets $1-4$ except with one difference. The ground truth RUL for the test set are not revealed. The challenge participants were asked to upload their results (only once per day) to receive a score based on an asymmetrical scoring function (see (Saxena, Goebel, et al., 2008a)). Users can still get their results evaluated using the same scoring function by uploading their results on the repository page, but otherwise it is not possible to compute any other metric on the results in absence of ground truth to allow error computation. The third part of the challenge set is dataset $\#5V$, the final *validation set* that was used to rank the challenge participants, where they were allowed only once chance to submit their results. The challenge since then is still continuing and a participant may submit final results (only once) for evaluation per instructions posted with the dataset on the NASA repository (Saxena & Goebel, 2008).

2.2. Performance Benchmarking

One of the key drivers for this study was to assess state-of-the-art in prognostic methods established through comparisons and performance benchmarking. However, the survey revealed a serious lack of consistency in methods used for

Table 1. Description of the five turbofan degradation datasets available from NASA repository.

Datasets		#Fault Modes	#Conditions	#Train Units	#Test Units
Turbofan data from NASA repository	$\#1$	1	1	100	100
	$\#2$	1	6	260	259
	$\#3$	2	1	100	100
	$\#4$	2	6	249	248
PHM2008 Data Challenge	$\#5T$	1	6	218	218
	$\#5V$	1	6	218	435

performance evaluation. One of the key contributing reasons towards this inconsistency is thought to be the unavailability of established performance banchmark. Originally it was planned that the PHM08 challenge winning performances would establish a benchmark that would allow further improvements as new methods are developed. But since that webpage was taken down in subsequent years these scores have not been easily available except as reported (often partially) in some publications from the winners. It is, therefore, planned to compute several relevant metrics on the submitted results during PHM08 challenge and make them available to serve as reference for future efforts. These benchmarks, however, remain beyond the scope of this paper and will be made available in future publications.

3. C-MAPSS DATASET LITERATURE REVIEW

To analyze various approaches that have been used to solve C-MAPSS dataset problem, all the publications that cite these datasets including the references recommended by the repository were collected through standard web search. The search results returned over seventy publications which were then preprocessed to identify overlapping efforts by same authors or the publications that only cite the dataset but perceivably did not use them for algorithm development. This resulted in forty unique publications that were then considered for review and analysis in this work.

For the sake of readability, each of these publications were assigned a unique ID to use in various tables summarizing the results presented in this section. This mapping between publication and IDs is presented in Table 10 as appendix. Furthermore, to keep the paper length short, a detailed review analysis of each of the forty publications is not included but only the summarized findings.

The analysis of the collected publications reveals several important observations that are summarized here. First, these publications are binned into various different categories and then analyzed for the distributions thus observed. These categories and corresponding findings are presented next.

3.1. C-MAPSS Dataset Used

Table 2 identifies specific publications that use one or more of these five datasets. It can be observed that the dataset $\#1$ was the most used one (55%), followed by the test set ($\#5T$) from the PHM08 challenge (35%), whereas rest of the other datasets are relatively under utilized. Three publications report generating their own datasets using the C-MAPSS simulator and (Richter, 2012) describes the simulator and how it can be used to generate degradation data rather than using any specific dataset.

The heavy usage of the dataset $\#1$ ($\approx$ 70%) compared to all other datasets among the four from the NASA Repository may be attributed to its apparent simplicity compared to the rest because some of the sensor measurements in this dataset depict a monotonic trend. This may lead to a possible confusion with health indicators. High usage of dataset $\#5T$ is attributed to the PHM08 challenge, where several teams had already used these data extensively, thereby gaining significant familiarity with the dataset as well as a preference due to availability of corresponding benchmark performance from the challenge leader board.

Table 2. List of publications for each dataset.

Datasets		Publication ID	Ratio
Turbofan data from NASA repository	$\#1$	5, 6, 10, 13, 14, 15, 19, 20, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 36, 37, 38, 40	22/40
	$\#2$	13, 22, 34, 40	4/40
	$\#3$	34, 40	2/40
	$\#4$	7, 34, 40	3/40
PHM08 Data challenge	$\#5T$	1, 2, 3, 4, 8, 12, 16, 17, 21, 29, 30, 34, 35, 40	14/40
	$\#5V$	1, 2, 3, 40	4/40
Simulator	OWN	9, 11, 39	3/40
Other	-	18	1/40

Several publications mentioned in Table 2 have used only the training datasets that have complete (run-to-failure) trajectories. Using data with complete trajectories gives access to the true End-of-Life (EOL) to compute RUL from any time point in a degradation trajectory which could be used to generate a larger set of training data. This approach is also relevant to estimating RUL at different time points and allows the usage of prognostics metrics (Saxena, Celaya, et al., 2008) such as Prognostic Horizon, $\alpha - \lambda$ metric, or the convergence measure. However, in true learning sense the algorithm, once trained, must be tested on unseen data for proper validation, as was required for the PHM'08 challenge

datasets. Table 3 shows that 11 different publications used the full training/testing datasets: the training dataset for estimating the parameters of the algorithms and using the full testing datasets for performance evaluation.

Table 3. List of publications using only *full* training/testing datasets.

Datasets		Publication ID	Ratio
Turbofan dataset from NASA repository	$\#1$	20, 27, 28, 40	5/40
	$\#2$	40	1/40
	$\#3$	40	1/40
	$\#4$	40	1/40
PHM08 Data challenge	$\#5T$	1, 2, 3, 4, 16, 21, 40	7/40
	$\#5V$	1, 2, 3, 40	4/40

3.2. Target Problem Being Solved

As normally expected there is a wide variety of approaches taken in interpreting the datasets, formulating a problem. and modeling the system to solve the problem. However, contrary to expectations a significant number of publications have utilized these datasets for analysis heavily focused on diagnosis (multi-class classification) rather than prognostics.

By posing a multi-class classification problem various publications attempt to solve mainly three types of problems:

- Supervised classification: The training dataset is labeled (known classes for each feature vector);
- Unsupervised classification: The classes are not known apriori and data are not labeled;
- Partially supervised classification: Some classes are precisely known, others are unknown or are attached with a confidence value to express belief in that class.

Publications 1, 7, 10, 20, 24, 27, 32 use classification for preprocessing steps towards solving a prognostics problem. Specifically, unsupervised classification algorithms are used in publications 1, 7 to segment the dataset into the six operating conditions. For reference, detailed information about various simulated operating conditions in C-MAPSS is described in (Richter, 2012), which can also be used to label these datasets. Supervised and unsupervised classification algorithms are also used in publications 6, 10, 20, 27, 32 to assign a degradation level according to sensor measurements. The sequence of discrete failure degradation stages is indeed relevant for the estimation of the current health state and its prediction (Kim, 2010).

Health assessment, anomaly detection (seen as a 1-class classification problem) or fault identification are tackled in publications 6, 11, 12, 13, 26, 31, 35 using supervised classification methods, and partially supervised classification techniques in publications 12, 27, 33. For these approaches, a known target (or a degradation level) is required to evaluate the classification rate. For instance, four degradation levels were defined for labeling data in publications 6, 10, 27, 33: normal degradation (class 1), knee corresponding to a noticeable degradation (class 2 viewed as a transition between class 1 and 3), accelerated degradation (class 3) and failure (class 4). One such segmentation is provided at URL[1], whereas a different set of segmentation was proposed in publication 13. Using these segmented data (clusters) as proxy to ground truth, some level of classification performance can be evaluated for comparison purposes.

Similar to several classification approaches used, many approaches were employed for solving the prognostics problem for predicting RUL. In order to give due attention to the analysis of prognostic methods, a discussion is presented separately in Section 4.

3.3. Method for Treatment of Uncertainty

Given the inherent nature of datasets that include several noise factors and lack of specific information on the effects of operational conditions it is important for algorithms to model and account for uncertainty in the system. Different publications have dealt with uncertainty at various stages of processing as described below:

1. **Signal processing step** such as noise filtering using a Kalman filter as in publications 2, 3, 20, Gaussian kernel smoothing in publications 1, 7, and functional principal component analysis in publication 15.
2. **Feature extraction/selection step** such as using principal component analysis and other variants of it as suggested in publications 1, 7, 13, grey-correlation in publication 22, and computing relevance of features for prediction in publication 23.
3. **Health estimation step** such as based on operating conditions assessment to normalize/factor out the effects of operating conditions as proposed in publications 1, 7, 21, 40 and using non-linear regression.
4. **Classification step** where uncertainty modeling plays a role on data labeling using noisy and imprecise degradation levels as shown in publications 12, 27, 33, or on the inference of a sequence of degradation levels such as using Markov Models or multi-models as in publications 6, 10, 24, 32, 34.
5. **Prediction step** such as gradually incorporating prior knowledge during estimation in presence of noise as proposed in publications 4, 14, 16, 17, 19, 21, 30, in determining failure thresholds as in publications 10, 27, 32 or in representing health indicator such as in publication 40 to be used in prediction.
6. **Information fusion step** by merging multiple RUL estimates through Bayesian updating as pointed in publica-

[1] http://members.femto-st.fr/emmanuel-ramasso/data-and-codes

tions 4, 21 or in similarity-based matching as in publications 1, 27, 40.

A variety of different uncertainty representation theories are found to be used. Table 4 classifies different publications according to the theory of uncertainty treatment used in corresponding analysis (Klir & Wierman, 1999). As shown in the table, the probability theory is the most popular one (65%) followed by set-membership approaches (in particular fuzzy-sets with 15%), Dempster-Shafer's theory of belief functions (13%), and other measures (such as polygon area and Choquet integral).

Table 4. Methods for uncertainty management used on C-MAPSS datasets.

Theories	Publication ID	Ratio
Probability theory	1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 15, 16, 17, 19, 20, 21, 22, 26, 28, 29, 30, 31, 32, 33, 34, 35	26/40
Set-membership	10, 14, 23, 25, 36, 39	6/40
Belief functions	6, 10, 24, 27, 33	5/40
Other measures	10, 40	2/40

3.4. Methods used for Performance Evaluation

Table 5 summarizes the performance measures that have been used for prognostics-oriented publications. A taxonomy of performance measures for RUL estimation was proposed in (Saxena, Celaya, et al., 2008; Saxena, Celaya, Saha, Saha, & Goebel, 2010), where different categories were presented: accuracy-based, precision-based, robustness-based, trajectory-based, computational performance and cost/benefit measures, as well as some measures dedicated specifically to prognostics (PHM metrics). Since this problem involves predictions on multiple units, it is expected that the majority of publications would use error-based accuracy and precision metrics. Metric like the Mean Squared Error (MSE) has been used in two different ways: For the estimation of the goodness of fit between a predicted and a real signal, and as an accuracy-based metric to aggregate errors in RUL estimation. Only the publications that fall under latter category are included in the table. The table clearly shows that accuracy-based measures were most widely used, in particular the scoring function from PHM08 challenge, which also weighs accuracy by timeliness of predictions. Broader usage of this metric is also explained by the fact that this is the only metric for which scores from data challenge were available and can be used as benchmark to compare with any new development. However, one may also compute additional measures if using only the training datasets where full trajectories are available. In that case, approaches like leave-one-out validation become applicable where all training instances but one are used for training each time and the remaining one is used for performance evaluation. Then the average of the performance measure is computed from all the runs. Publication 27 presents this approach for dataset $\#1$ and a cross-validation procedure for dataset $\#5T$ is used in publication 21. Note that publications 19, 20, 32 provide the only RUL estimates for all testing instances (without computing any metrics) and publications 10, 27 present distribution of errors.

Table 5. Performance measures used in prognostics-oriented publications applied on C-MAPSS.

Categories	Measures	Publication ID	Ratio
Accuracy	PHM08 Score	1, 2, 4, 5, 8, 16, 21, 29, 30, 40	10/40
	FPR, FNR	8, 10, 27, 40	4/40
	MSE	3, 8, 15, 17, 29, 40	6/40
	MAPE	4, 23, 28, 32, 34, 39, 40	7/40
	MAE	5, 13, 38, 40	4/40
Precision	ME	25,28,32,39	4/40
	MAD	25	1/40
Prognostics	PH	7, 22	2/40
	$\alpha - \lambda$	7, 22	2/40
	RA	7, 22, 34	3/40
	CV	7, 22, 34	3/40
	AB	34	1/40

4. PROGNOSTIC APPROACHES

C-MAPSS datasets were generated to allow development and benchmarking of various prognostics approaches. However, as observed from the literature review (see Section 3.2) many researchers have used them to cast a multiclass classification problem instead, even though majority of publications did use them to develop prognostics algorithm. This section focuses on describing those prognostic approaches. These approaches used on C-MAPSS datasets can be divided into three broad categories as described next.

4.1. Category 1: Using functional mappings between set of inputs and RUL

Methods in this category (see Table 6) first transform the training data (trajectories) into a multidimensional feature space and use corresponding RUL to label corresponding feature vectors. Then using supervised learning methods a mapping between feature vectors and RUL is developed. Methods within this category are mostly based on Neural Networks with various architectures. Different sensor channels were used to generate corresponding features. However, it was observed that the approaches yielding good performance also included a feature selection step through advanced parameter optimization such as using genetic algorithm and Kalman filtering as described in publications 2, 3 that ranked 2d and 3rd respectively in the competition.

4.2. Category 2: Functional mapping between health index (HI) and RUL

Methods listed in Table 7 are based on the estimation of two mapping functions: One maps sensor measurements to

Table 6. Category 1 methods using a mapping learned between a subset of sensor measurements as inputs and RUL as output.

Methods	Publication ID
RNN, EKF	2
MLP, RBF, KF, Ensemble	3
MLP	8
ANN	9
ESN	20
Fuzzy rules, genetic algorithm	36
MLP, adaboost	38

a health index (1-D variable) for each training unit based on sensor measurements; The second mapping links health index values to the RUL. These approaches construct a library of degradation models. Inference of the RUL for a given test instance includes using the library as prior knowledge to update the parameters of the model corresponding to the new test instance. Updating can be done using Bayes rule as proposed in publication 4 or other model averaging or ensemble techniques designed to take into account the uncertainty inherent to the model selection process (Raftery, Gneiting, Balabdaoui, & Polakowski, 2003).

Table 7. Type 2 methods using health index as input and RUL as output.

Methods	Publication ID
Quadratic fit, Bayesian updating	4
Logistic regression	5
Kernel regression, RVM	7
RVM	16
Gamma process	17
Linear, Bayesian updating	19
RVM, SVM, RNN, Exponential and quadratic fit, Bayesian updating	21
Exponential fit	28
Wiener process	29
Copula	30
HMM, LS-SVR	34

Table 8 lists some other approaches that use approximation functions to represent the evolution of individual sensor measurement through time. Given a test instance as many predictions are made as the number of sensors. These predictions are then used in a classifier that assigns a class label related to identified degradation level. Some of these approaches also update classifier parameters with new measurements using some Bayesian updating rules as mentioned previously. These methods were however applied only on dataset #1 in which sensors depict clear monotonic trends.

4.3. Category 3: Similarity-based matching

In these methods (Table 9), historical instances of the system (sensor measurements trajectories labeled with known failure times) are used to create a library. For a given test instance

Table 8. Category 2 methods based on individual sensor modeling and classification.

Methods	Publication ID
exTS, supervised classification	10
SVR	13
exTS, ARX	14
ANN, ANFIS	23
Piece-wise linear (multi-models)	24
exTS	25
ELM, unsupervised classification	32

similarity with instances in the library is evaluated generating a set of Remaining Useful Life (RUL) estimates that are eventually aggregated using different methods. Compared to category 2 methods, these methods do not make use of training trajectory abstraction into features, but trajectory data (possibly filtered) are themselves stored. Similarity is computed in the sensor space as in publication 27 or using health indices as in publications 1, 7, 17, 21, 40.

As mentioned in publications 1, 7, in practice, the test instance and the training instance may take different time in reaching a particular degradation level from the initial healthy state. Therefore, similarity-based matching must accommodate this difference in the early phases of degradation curves. In publication 40, this problem was tackled by assuming a constant initial wear for all instances yielding an offset on health indices. Efficient similarity measures are also necessary to cope with noise and degradation paths. For instance, in publications 1, 7 three different similarity measures were used, and in publication 40, computational geometry tools were used for instance representation and similarity evaluation.

Table 9. Category 3 methods using similarity-based matching.

Methods	Publication ID
HI-based 3 similarity measures and kernel smoothing	1, 7
Similar to 1 and 7 using 1 similarity measure	22
Feature-based similarity, 1 similarity measure, ensemble, degradation levels classification	27
HI-based similarity, polygon coverage similarity, ensemble	40

An advantage of approaches in this category is that new instances can be easily incorporated. Moreover, similarity-based matching approaches have demonstrated good generalization capability on all C-MAPSS datasets as shown in publications 1, 7, 40 despite a high level of noise, multiple simultaneous fault modes, and a number of operating conditions. This category of algorithms are relatively easily parallelized to reduce computational times needed for inference.

5. SOME GUIDELINES TO USING C-MAPSS DATASETS

Another contribution from this paper is through summarizing some guidelines in using C-MAPSS datasets that my help future users to understand and utilize these datasets better. It summarizes information gathered from the literature review and authors' own experiences, which in many cases goes beyond the documentation provided along with the datasets. Specifically, it offers some general processing steps and lists relevant publications that describe implementation of these preprocessing steps that could be useful in developing a prognostic algorithm (Figure 1).

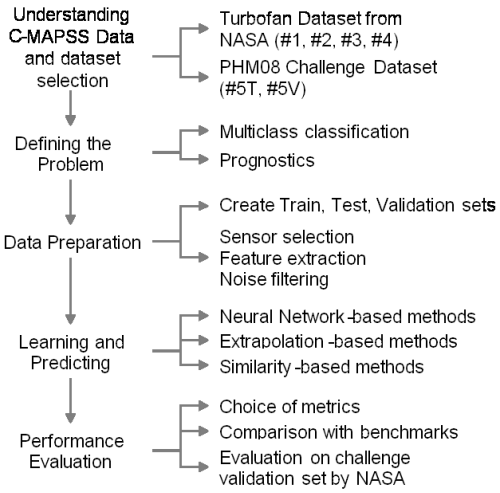


Figure 1. Guidelines to Using C-MAPSS Datasets.

Based on the analysis presented in (Section 3), five general data processing and algorithmic steps are considered:

[Step 1:] Understanding C-MAPSS datasets – Comprehensive background information on turbofan engines and C-MAPSS datasets is well presented in three publications, (Saxena, Goebel, Simon, & Eklund, 2008b), (Richter, 2012), and (T. Wang, 2010). More details about the hierarchical decomposition of the simulated system into critical components can also be found in (Frederick et al., 2007; Abbas, 2010), which provides valuable domain knowledge. These publications do not focus on the physics-of-failure of turbofan engines but describe generation of these datasets and various practical aspects when using C-MAPSS datasets for prognostics. These include description of sensors measurements, illustrations of operating conditions, impact of fault modes, etc., which can play an important role in improving data-driven prognostics algorithms as well. Going from dataset $\#1$ to $\#4$ represents varying degrees of complexity and, therefore, it is recommended to use them in that order to incrementally develop methods to accommodating individual complexity one by one. The challenge datasets fall somewhere in the middle as far as complexity level goes but suffer from availability of ground truth information for a quicker feedback during algorithm development. Therefore, these datasets may be used as validation examples and should be compared to other approaches using benchmarks presented in Section 2.2.

[Step 2:] Defining the problem – Given the nature of these datasets several types of problems can be defined. As mentioned in Section 3.2 in addition to prediction, a multi-class classification problem can be defined for a multidimensional feature space. However, the intent behind these data was to promote prognostics algorithm development. Since these data consist of multiple trajectories, the problem to predict the RUL for all trajectories can be constructed just as the one posed in the data challenge. However, one could also define the problem at a higher granularity by modeling the degradation for each trajectory individually and predict RUL at multiple time instances, which would be more of a condition based prognostics context.

[Step 3:] Data preparation – After a dataset (turbofan or data challenge) is selected, it is suggested to split the original training dataset into two subsets: a training dataset for model parameter estimation (learning) and a testing dataset to test the learned model 7 (see for example publications 21, 40). For the datasets $\#1 - 4$ corresponding RUL vectors are provided for the test sets so users can validate their algorithms. However, for the challenge datasets, the evaluations can only be obtained by uploading the RUL to the data repository website. Therefore, it may be desirable to split the training set itself for training, test, and validation purposes during algorithm development. The next step is to downselect sensors to reduce problem dimensionality. Some data exploration and preparation approaches for the data challenge (datasets $\#5T$ and $\#5V$) are well described in publications 1, 2 and 7. Some "heuristic rules" to avoid over-predictions are also presented in publication 40 and applied on all five C-MAPSS datasets. Some of the better performing methods are based on a PCA such as in publication 1, and other sensor selection procedures such as in publications 2, 3 and 40. From the survey it was noted that the most commonly selected subset of sensors was $7, 8, 9, 12, 16, 17, 20$ (as it was also initially suggested in publication 1). Additional sensors may also be considered, similar to the approach proposed in publication 40 where a total of 511 combinations were studied for each dataset for an exhaustive evaluation.

[Step 4:] Learning and Predicting – This step forms the core of the prediction problem. As described in Section 3 a variety of learning approaches can be employed to learn various mappings between the sensor data and system health to compute RUL. Some of these methods try to learn RUL as a function of sensor data (system state) or features thereof, others estimate a health index first. Each of the trajectory can be modeled into a degradation process to predict when they cross the zero health threshold using regression methods. Ap-

proaches based on health index computation can be applied to all datasets. The approach proposed in publications 1, 7 is the simplest to implement. To deal with normalization (or alternatively segmentation) of data by operating conditions one could use a clustering approach as suggested by the authors above, or one may directly use the parameters described in publication 18 to validate the performance of segmentation. Some variants for health indicator estimation can also be picked from publications 21 and 40.

[Step 5:] Performance evaluation – Once a learned model results in to satisfactory results on the testing set aside by partitioning the training data, one may use the actual test dataset provided with the datasets. After further tuning, especially for datasets ($\#5T$ and $\#5V$), a final validation can be done by submitting the results to the NASA repository. Before uploading the final submission, the generalization capability should be ensured by computing using several performance metrics as discussed in Section 2.2. Some benchmarks have been provided in Section 2.2 using metrics that aggregate prediction performance from multiple units. While the exact numbers would not match, the performance is expected to be in the similar range for results obtained from turbofan datasets that have access to RUL. For comparison purposes, the scores obtained in previous works on complete C-MAPSS trajectories are summarized in publication 40. Note that here using the full trajectory data it is possible to compute prognostics metrics as presented in (Saxena, Celaya, et al., 2008; Saxena et al., 2010) as the actual EOL is known apriori. This allows testing the critical time aspect of a prediction in addition to accuracy and precision measures.

6. CONCLUSION

As observed from published PHM literature the most widely used datasets for data-driven prognostics come from the C-MAPSS turbofan simulator from among the other openly available prognostic datasets. Guided by this observation, a survey of approaches developed using these datasets (since 2008) was carried out with the purpose of understanding the current state-of-the-art and assess how these datasets have helped in development of prognostic algorithms. However, it was noticed that due to several factors, these datasets did not get used as intended and any meaningful comparison between approaches was not trivial. Specifically following observations were made and this paper tries to alleviate some of these factors to improve usage of these datasets as originally intended.

- Despite several thousand downloads only 70 papers referring to C-MAPSS were found in the published literature. This suggests that a vast majority of those who downloaded did not get to utilize these data to the point of publishing the results in a publication. Therefore, some guidance has been provided to help in understanding these datasets and how a prognostics problem may be set up in few different ways. Furthermore, a description of all five C-MAPSS datasets is provided identifying their distinguishing characteristics and clearing up some misunderstandings as identified from the survey.
- Among the 70 papers, only a few actually used the testing datasets for evaluating their methods. A mix of different datasets and the metrics used to evaluate performance was observed from the survey. This made it difficult to compare performance between different reported methods in a consistent manner. Therefore, a better explanation of differences in these datasets and providing the top thirty scores from challenge datasets should help future users in comparing their methods against a benchmark in a more consistent manner. Furthermore, it is also suggested how results from datasets that are not from the challenge could be compared against this benchmark established on the challenge set.
- The survey reveals usage of various prognostics approaches that can be divided into three main categories. These approaches are briefly described with potential areas for further improvement. The survey also demonstrated that C-MAPSS datasets can be used for developing and testing methods for several intermediate steps in prognostics such as sensor selection, health indicator estimation, operating conditions modeling in addition to fault estimation and prediction.

With the analysis presented in this paper and references to a variety of approaches employed, this paper hopes to establish public knowledge that can be used by future users in prognostic algorithm development and aid in fulfilling the underlying intent of data repository to facilitate algorithm benchmarking and further development. The issue of performance benchmarking remains to be explored as part of future work where authors plan to compute performance for challenge entries based on several other metrics that will allow comparisons with performance results reported in many publications.

NOMENCLATURE

PHM	Prognostics and Health Management
RUL	Remaining Useful Life
CMAPSS	Commercial Modular Aero-Propulsion System Simulation
HI	Health index
MLP	MultiLayer Perceptron
ANN	Artificial neural network
RNN	Recurrent neural network
RBF	Radial basis function
ESN	Echo state network
ELM	Extreme learning machine
EKF	Extended Kalman filter

KF	Kalman filter
SVR	Support vector regression
LS-SVR	Least squared support vector regression
exTS	Evolving extended Takagi-Sugeno system
ARX	Autoregressive exogeneous model
ANFIS	Adaptive neuro fuzzy inference system
RVM	Relevance vector machine
HMM	Hidden Markov model
PCA	Principal components analysis
MSE	Mean squared error
MAPE	Mean absolute percentage error
MAE	Mean absolute error
ME	Mean error
PH	Prediction horizon
AP	Acceptable predictions (rate)
$\alpha - \lambda$	Accuracy at specific times
RA	Relative accuracy
CV	Convergence
AB	Average bias
FPR	False positive rate
FNR	False negative rate

ACKNOWLEDGMENT

This work has been partly carried out within the Laboratory of Excellence ACTION through the program "Investments for the future" managed by the French National Agency for Research (ANR-11-LABX-01-01) and with partial support from NASA's System-wide Safety and Assurance Technologies (SSAT) Project under ARMD/Aviation Safety program.

REFERENCES

Abbas, M. (2010). *System level health assessment of complex engineered processes*. Unpublished doctoral dissertation, Georgia Institute of Technology.

Al-Salah, T., Zein-Sabatto, S., & Bodruzzaman, M. (2012). Decision fusion software system for turbine engine fault diagnostics. In *Southeastcon, 2012 proceedings of ieee* (p. 1-6).

Coble, J. (2010). *Merging data sources to predict remaining useful life - an automated method to identify prognostic parameters*. Unpublished doctoral dissertation, University of Tennessee, Knoxville.

Coble, J., & Hines, J. (2008). Prognostic algorithm categorization with phm challenge application. In *Ieee int. conf. on prognostics and health management.*

Coble, J., & Hines, W. (2011). Applying the general path model to estimation of remaining useful life. *International Journal of Prognostics and Health Management*, *2*, 1-13.

El-Koujok, M., Gouriveau, R., & Zerhouni, N. (2011). Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system. *Microelectronics Reliability*, *51*(2), 310 - 320.

Frederick, D., DeCastro, J., & Litt, J. (2007). *User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS)* (Tech. Rep.). Cleveland, Ohio 44135, USA: National Aeronautics and Space Administration (NASA), Glenn Research Center.

Gouriveau, R., Ramasso, E., & Zerhouni, N. (2013). Strategies to face imbalanced and unlabelled data in PHM applications. *Chemical Engineering Transactions*, *33*, 115-120.

Gouriveau, R., & Zerhouni, N. (2012). Connexionist-systems-based long term prediction approaches for prognostics. *IEEE Trans. on Reliability*, *61*, 909-920.

Heimes, F. (2008). Recurrent neural networks for remaining useful life estimation. In *Ieee int. conf. on prognostics and health management.*

Hu, C., Youn, B., Wang, P., & Yoon, J. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering and System Safety*, *103*, 120 - 135.

Ishibashi, R., & Nascimento Junior, C. (2013). GFRBS-PHM: A genetic fuzzy rule-based system for phm with improved interpretability. In *Ieee conference on prognostics and health management (phm)* (p. 1-7).

Javed, K., Gouriveau, R., Zemouri, R., & Zerhouni, N. (2012). Features selection procedure for prognostics: An approach based on predictability. In *8th ifac symposium on fault detection, supervision and safety of technical processes* (Vol. 8, p. 25-30).

Javed, K., Gouriveau, R., & Zerhouni, N. (2013). Novel failure prognostics approach with dynamic thresholds for machine degradation. In *Ieee industrial electronics conference.*

Jianzhong, S., Hongfu, Z., Haibin, Y., & Pecht, M. (2010). Study of ensemble learning-based fusion prognostics. In *Prognostics and health management conference, 2010. phm '10.* (p. 1-7).

Kim, H.-E. (2010). *Machine prognostics using health state probability estimation*. Unpublished doctoral dissertation, School of engineering systems, Faculty of built environmental engineering, Queensland university of technology.

Klir, G., & Wierman, M. (1999). Uncertainty-based information. elements of generalized information theory. In (chap. Studies in fuzzyness and soft computing). Physica-Verlag.

Li, X., Qian, J., & Wang, G. (2013). Fault prognostic based on hybrid method of state judgment and regression. *Advances in Mechanical Engineering*, *2013*(149562), 1-10.

Liao, H., & Sun, J. (2011). Nonparametric and semi-parametric sensor recovery in multichannel condition monitoring systems. *IEEE Transactions on Automa-*

tion Science and Engineering, *8*(4), 744-753.

Lin, Y., Chen, M., & Zhou, D. (2013). Online probabilistic operational safety assessment of multi-mode engineering systems using Bayesian methods. *Reliability Engineering & System Safety*, *119*(0), 150 - 157.

Liu, K., Gebraeel, N. Z., & Shi, J. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Trans. on Automation Science and Engineering*.

Peel, L. (2008). Data driven prognostics using a Kalman filter ensemble of neural network models. In *Int. conf. on prognostics and health management.*

Peng, Y., Wang, H., Wang, J., Liu, D., & Peng, X. (2012). A modified echo state network based remaining useful life estimation approach. In *Ieee phm conference.*

Peng, Y., Xu, Y., Liu, D., & Peng, X. (2012). Sensor selection with grey correlation analysis for remaining useful life evaluation. In *Annual conference of the phm society.*

Raftery, A., Gneiting, T., Balabdaoui, F., & Polakowski, M. (2003). Using bayesian model averaging to calibrate forecast ensembles. *American Meteorological Society*, *133*(5), 1155-1174.

Ramasso, E. (2009). Contribution of belief functions to hidden Markov models with an application to fault diagnosis. In *Machine learning for signal processing.*

Ramasso, E. (2014a). Investigating computational geometry for failure prognostics. *Int. Journal on Prognostics and Health Management*. (submitted)

Ramasso, E. (2014b). Investigating computational geometry for failure prognostics in presence of imprecise health indicator: Results and comparisons on c-mapss datasets. In *European conf. on prognostics and health management.*

Ramasso, E., & Denoeux, T. (2013). Making use of partial knowledge about hidden states in hidden Markov models: an approach based on belief functions. *IEEE Transactions on Fuzzy Systems*, *10.1109/TFUZZ.2013.2259496*.

Ramasso, E., & Gouriveau, R. (2010). Prognostics in switching systems: Evidential Markovian classification of real-time neuro-fuzzy predictions. In *Ieee prognostics and health management conference.*

Ramasso, E., & Gouriveau, R. (2013). RUL estimation by classification of predictions: an approach based on a neuro-fuzzy system and theory of belief functions. *IEEE Transactions on Reliability*, *Accepted*.

Ramasso, E., Rombaut, M., & Zerhouni, N. (2013). Joint prediction of observations and states in time-series based on belief functions. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, *43*, 37-50.

Riad, A., Elminir, H., & Elattar, H. (2010). Evaluation of neural networks in the subject of prognostics as compared to linear regression model. *International Journal of Engineering & Technology*, *10*, 52-58.

Richter, H. (2012). Engine models and simulation tools. In *Advanced control of turbofan engines* (p. 19-33). Springer New York.

Sarkar, S., Jin, X., & Ray, A. (2011). Data-driven fault detection in aircraft engines with noisy sensor measurements. *Journal of Engineering for Gas Turbines and Power*, *133*, 081602.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, W. (2008). Metrics for evaluating performance of prognostic techniques. In *Int. conf. on prognostics and health management.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*.

Saxena, A., & Goebel, K. (2008). C-mapss data set. *NASA Ames Prognostics Data Repository*.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008a). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–9).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008b). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Int. conf. on prognostics and health management* (p. 1-9). Denver, CO, USA.

Serir, L., Ramasso, E., & Zerhouni, N. (2012). An evidential evolving multimodeling approach for systems behavior prediction. In *Annual conference of the phm society.*

Siegel, D. (2009). *Evaluation of health assessment techniques for rotating machinery*. Unpublished master's thesis, Division of Research and Advanced Studies of the University of Cincinnati.

Son, K. L., Fouladirad, M., & Barros, A. (2012). Remaining useful life estimation on the non-homogenous gamma with noise deterioration based on gibbs filtering : A case study. In *Ieee int. conf. on prognostics and health management.*

Son, K. L., Fouladirad, M., Barros, A., Levrat, E., & Iung, B. (2013). Remaining useful life estimation based on stochastic deterioration models: A comparative study. *Reliability Engineering and System Safety*, *112*, 165 - 175.

Sun, J., Zuo, H., Wang, W., & Pecht, M. (2012). Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance. *Mechanical Systems and Signal Processing*, *28*, 585 - 596.

Tamilselvan, P., & Wang, P. (2013). Failure diagnosis using deep belief learning based health state classification. *Reliability Engineering and System Safety*, *115*(0), 124 - 135.

Wang, P., Youn, B., & Hu, C. (2012). A generic probabilistic framework for structural health prognostics and uncer-

tainty management. *Mechanical Systems and Signal Processing*, *28*, 622 - 637.

Wang, T. (2010). *Trajectory similarity based prediction for remaining useful life estimation*. Unpublished doctoral dissertation, University of Cincinnati.

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Int. conf. on prognostics and health management* (p. 1-6).

Xi, Z., Jing, R., Wang, P., & Hu, C. (2013). A copula-based sampling method for data-driven prognostics and health management. In *Asme 2013 international design engineering technical conferences and computers and information in engineering conference.*

Xue, Y., Williams, D., & Qiu, H. (2011). Classification with imperfect labels for fault prediction. In *Proceedings of the first international workshop on data mining for service and maintenance* (pp. 12–16). ACM.

Yu, J. (2013). A nonlinear probabilistic method and contribution analysis for machine condition monitoring. *Mechanical Systems and Signal Processing*, *37*, 293-314.

Zein-Sabatto, S., Bodruzzaman, J., & Mikhail, M. (2013). Statistical approach to online prognostics of turbine engine components. In *Southeastcon, 2013 proceedings of ieee* (p. 1-6).

Zhao, D., P., R. G., & Willett. (2011). Comparison of data reduction techniques based on SVM classifier and SVR performance. In *Proc. spie, signal and data processing of small targets* (Vol. 8137, p. 1-15).

APPENDIX

All references were mapped to numeric identifiers to be used in survey and analysis results for better readability. This mapping is provided in the Table 10 below.

Table 10. References to ID mapping.

Reference	Publication ID
(T. Wang, Yu, Siegel, & Lee, 2008)	1
(Heimes, 2008)	2
(Peel, 2008)	3
(Coble & Hines, 2008) (Coble, 2010) (Coble & Hines, 2011)	4
(Siegel, 2009)	5
(Ramasso, 2009)	6
(T. Wang, 2010)	7
(Riad, Elminir, & Elattar, 2010)	8
(Abbas, 2010)	9
(Ramasso & Gouriveau, 2010) (Ramasso & Gouriveau, 2013)	10
(Sarkar, Jin, & Ray, 2011)	11
(Xue, Williams, & Qiu, 2011)	12
(Zhao, P., & Willett, 2011)	13
(El-Koujok, Gouriveau, & Zerhouni, 2011)	14
(Liao & Sun, 2011)	15
(P. Wang, Youn, & Hu, 2012)	16
(Son, Fouladirad, & Barros, 2012)	17
(Richter, 2012)	18
(Sun, Zuo, Wang, & Pecht, 2012)	19
(Peng, Wang, Wang, Liu, & Peng, 2012)	20
(Hu, Youn, Wang, & Yoon, 2012)	21
(Peng, Xu, Liu, & Peng, 2012)	22
(Javed, Gouriveau, Zemouri, & Zerhouni, 2012)	23
(Serir, Ramasso, & Zerhouni, 2012)	24
(Gouriveau & Zerhouni, 2012)	25
(Yu, 2013)	26
(Ramasso, Rombaut, & Zerhouni, 2013)	27
(Liu, Gebraeel, & Shi, 2013)	28
(Son, Fouladirad, Barros, Levrat, & Iung, 2013)	29
(Xi, Jing, Wang, & Hu, 2013)	30
(Lin, Chen, & Zhou, 2013)	31
(Javed, Gouriveau, & Zerhouni, 2013)	32
(Ramasso & Denoeux, 2013)	33
(Li, Qian, & Wang, 2013)	34
(Tamilselvan & Wang, 2013)	35
(Ishibashi & Nascimento Junior, 2013)	36
(Gouriveau, Ramasso, & Zerhouni, 2013)	37
(Jianzhong, Hongfu, Haibin, & Pecht, 2010)	38
(Zein-Sabatto, Bodruzzaman, & Mikhail, 2013) (Al-Salah, Zein-Sabatto, & Bodruzzaman, 2012)	39
(Ramasso, 2014b) (Ramasso, 2014a)	40

Enhanced Trajectory Based Similarity Prediction with Uncertainty Quantification

Jack Lam[1], Shankar Sankararaman[2], and Bryan Stewart[3]

[1,3]*Naval Surface Warfare Center Port Hueneme Division, Port Hueneme, CA, 93043, USA*
jack.lam@navy.mil
bryan.stewart2@navy.mil

[2]*SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov

ABSTRACT

Today, data driven prognostics acquires historic data to generate degradation path and estimate the Remaining Useful Life (RUL) of a system. A successful methodology, Trajectory Similarity Based Prediction (TSBP) that details the process of predicting the system RUL and evaluating the performance metrics of the estimate was proposed in 2008. Two essential components of TSBP identified for potential improvement include 1) a distance or similarity measure that is capable of determining which degradation model the testing data is most similar to and 2) computation of uncertainty in the remaining useful life prediction, instead of a point estimate. In this paper, the Trajectory Based Similarity Prediction approach is evaluated to include Similarity Linear Regression (SLR) based on Pearson Correlation and Dynamic Time Warping (DTW) for determining the degradation models that are most similar to the testing data. A computational approach for uncertainty quantification is implemented using the principle of weighted kernel density estimation in order to quantify the uncertainty in the remaining useful life prediction. The revised approach is measured against the same dataset and performance metrics evaluation method used in the original TBSP approach. The result is documented and discussed in the paper. Future research is expected to augment TSBP methodology with higher accuracy and stronger anticipation of uncertainty quantification.

1. INTRODUCTION

Data driven prognostics acquires historic data to generate degradation path and estimate the Remaining Useful Life (RUL) of a system. In 2008, a new approached known as the Trajectory Similarity Based Prediction (TSBP) methodology was proposed in (Wang T. , 2013), and was successfully demonstrated during the NASA AMES 2008 Prognostics Health Management (PHM) challenge by obtaining the highest score by using a data-driven prognostics method to predict the RUL of a turbofan engine (Saxena & Goebel, PHM08 Challenge Data Description, 2008). While the TSBP is a proven technique, (Wang T. , 2013) does not address imbalanced data (Gouriveau, Ramasso, & Zerhouni, 2013), the effectiveness of different dissimilarity measure (Giusti, 2013), and uncertainty of the model (Dallachiesa, Nushi, Mirylenka, & Palpanas, 2012). These considerations are required to minimize the variation that exists in the data driven prognostics method, and systematically quantify the uncertainty in the RUL prediction.

In (Wang T. , 2013), the author developed a novel RUL prediction method based on the Instance Based Learning methodology called TSBP. In TSBP, the historical instances of a system with life-time condition data and known failure time from the training data are used to create a library of degradation models; these models are then compared against the testing data in order to compute a similarity measure and predict an RUL corresponding to each of the degradation models. The final RUL estimate can be obtained by aggregating the multiple RUL estimates using a density estimation method. While (Wang T. , 2013) focused on the basic TSBP methodology, there are still several areas for improvement.

For example, in (Yu, Yong, Datong, & Xiyuan, 2012), the authors investigated sensor selection as a critical research topic for prognostics. In their research, the authors stated that inclusion of irrelevant or redundant variables during data fusion may lead to over-fitting or less sensitivity of prognostics model, which would lead to adverse prediction performance.

Jack Lam et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In (Guo, Gerokostopoulos, Liao, & Niu, 2013), the authors proposed to incorporate degradation initiation time into the general degradation path modeling. Their paper argued that there is a "degradation free" period, i.e., degradation starts only after an initiation time and that a product failure is a combined effect of the initiation time and the degradation growth. In (Gouriveau, Ramasso, & Zerhouni, 2013), the authors suggested the need to deal with 1) data whose relative number of instances in each class evolves with time and 2) data whose significance is not known by the user. In (Giusti, 2013)and (Otey & Parthasarathy, 2004), both authors examined the notion of quantifying the dissimilarity between different multivariate time series. Their argument suggested that calculating the Euclidean distance between the centroids of two data sets is ineffective because it ignores the correlations present in the data sets. Finally, in (Dallachiesa, Nushi, Mirylenka, & Palpanas, 2012), the authors summarized the uncertainty in time series and suggested two main approaches to model these uncertain time series. Given all these factors, it can be easily seen that the TSBP method proposed in (Wang T. , 2013) can be reviewed and be improved.

In (Lei & Govindaraju, 2004), authors proposed the use of Simple Linear Regression (SLR) as a similarity measure technique for on-line signature recognition applications in comparison with the traditional approach of computing the Euclidean Distance, while having lower time complexity (O(n)) than Dynamic Time Warping (DTW) ($O(n^2)$). The SLR method utilized the mean-deviation normalization to circumvent the problem of scaling and shifting, which, in general, impacts the performance of the DTW method. Further, SLR can be adapted to multi-dimensional sequences, where most real-life applications are relevant.

In this paper, we examine the use of SLR and DTW within the TSBP method for similarity prediction and address the various shortcomings of the original TBSP approach that were explained in the previous paragraphs. Further, we test the result on the original dataset (Saxena & Goebel, 2008) and use the original performance evaluation metrics (Saxena, Celaya, Saha, Saha, & Goebel, 2009) against the original TBSP approach described by (Wang T. , 2013). We also compare the results using different density estimation approaches. The TSBP method with SLR and DTW as the similarity measure with the use of the kernel density estimation provide us with more insight into the problem.

The motivation for this work is to improve further the TSBP method by incorporating different similarity measures and develop a better understanding for uncertainty qualification. Although more work is needed to compare the results of TBSP methodology against the state-of-the-art data driven technique used by the industry, our study produced a survey of related areas that can be experimented to serve as an improved TSBP method. The target application is highly complex systems where physical modeling will be difficult and state of the operating condition can be observed. In this case, TSBP method can generate different degradation models against each regime from the different operating condition to generate an aggregation of RUL estimation. Unlike (Wang T. , 2013), this paper 1) anticipates imbalanced data, 2) evaluates the SLR and DTW similarity measures, and 3) incorporates the uncertainty modeling done in (Dallachiesa, Nushi, Mirylenka, & Palpanas, 2012). These capabilities further support the practical feasibility of the proposed method used in real applications. We envision more interest and study in the TBSP approach will drive academic community and industry into maturing the methodology to provide more accurate RUL estimation.

The rest of this paper is organized as follows. In Section 2, we review the multi-regime partitioning and normalization method used in (Wang T. , 2013). In Section 3, we briefly review the techniques for degradation modeling explained in (Wang & Coit, 2007) and (Guo, Gerokostopoulos, Liao, & Niu, 2013). In Section 4, we describe the similarity/dissimilarity measure used in (Dallachiesa, Nushi, Mirylenka, & Palpanas, 2012), (Yu, Yong, Datong, & Xiyuan, 2012), (Giusti, 2013), (Otey & Parthasarathy, 2004), and (Lei & Govindaraju, 2004). In Section 5, we describe uncertainty quantification in RUL estimation and review the density estimation methods. In Section 6, we include the discussion of the performance metrics described in (Saxena, et al., 2008). In Section 7, we review the dataset (Saxena & Goebel, 2008) and describe the procedures for the experiment. In Section 8 and 9 we present results and findings then conclude the paper in Section 10.

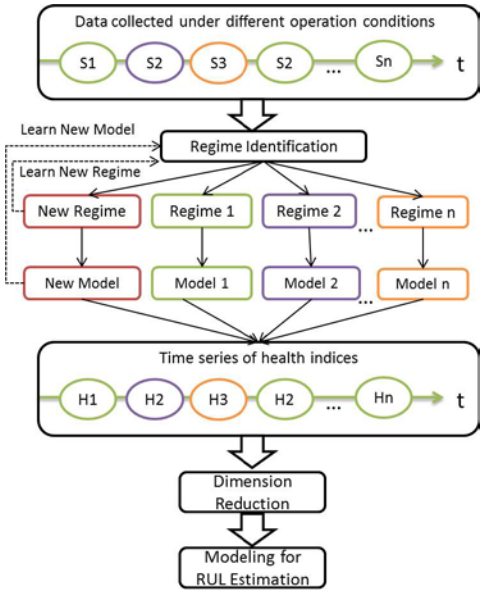


Figure 1. Process for multi-regime health assessment.

2. MULTI-REGIME PARTITIONING AND NORMALIZATION

When a system is operating under multiple operating conditions, the sensor measurements can behave differently in those unique environments, thereby causing difficulty in identifying failure trends. It is beneficial to identify the unique operating conditions or regimes from which sensors can be normalized or features can be extracted. Figure 1shows the high level process for multi-regime health assessment.

To illustrate multi-regime partitioning, the "Turbofan Engine Degradation simulation" data set from (Saxena & Goebel, PHM08 Challenge Data Description, 2008) will be examined. Within this data set, there are 21 sensor measurements and three other measurements that describe the operational conditions the system was operated under. The operating conditions change for each measurement (cycle). Figure 2 shows a select number of sensor measurements for the life time of one particular system.

2.1. Regime Identification

The first step in the process for multi-regime health assessment is to identify the unique, non-overlapping regimes. In this paper, multiple regimes are found using *k*-means clustering. The *k*-means clustering algorithm finds the optimum number of clusters, *k*, where each observation belongs to the nearest cluster's mean, hence the name *k*-means. Figure 3 shows the results of the *k*-means clustering algorithm on the "Turbofan Engine Degradation simulation" data set. As seen in Figure 3, the data was found to have 6 nicely separated and non-overlapping regimes.

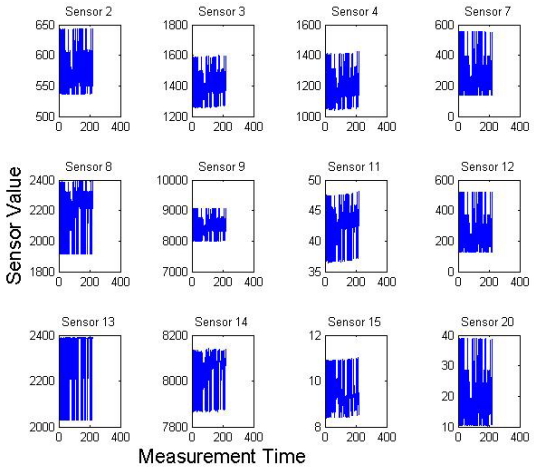


Figure 2. sensor measurement from "Turbofan Engine Degradation simulation" data set.

2.2. Mean-Variance Normalization

The next step is to normalize the sensor data according to the regime the measurement was taken under. This is done by performing mean-variance normalization. Similar to Eq. (1) where p represents the regime the sensor measurement belongs at time instance i.

$$y_i = \frac{x_i{}^p - \mu^p}{\sigma^p} \quad (1)$$

The mean-variance normalized data becomes the time series health indices as depicted in Figure 1. In continuation of the illustration, the progression from Figure 2 to Figure 4 shows a more revealing portrayal of the system behavior once the operating conditions are taken into consideration.

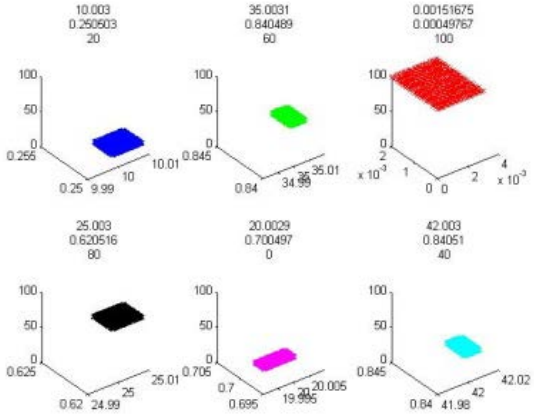


Figure 3. Multi-regime partitioning of the "Turbofan Engine Degradation simulation" data set. This figure represents all the operational condition that was performed.

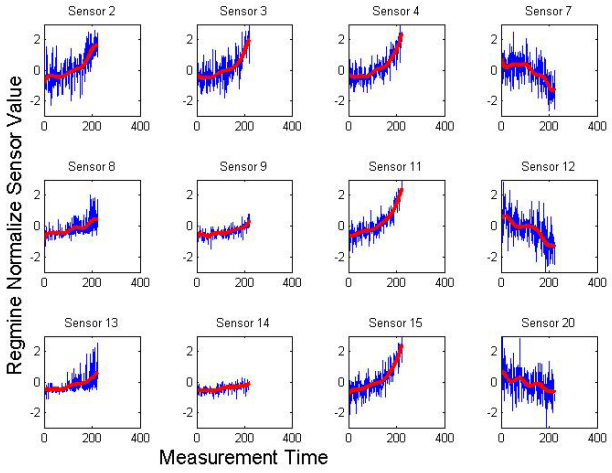


Figure 4. Mean variance normalized data (blue line) from the "Turbofan Engine Degradation simulation" data set for a single system or unit. The red line shows the degradation model for each sensor.

2.3. Variable Weighting and Dimensionality Reduction

At this point, the system data has been prepared and normalized for training the degradation models. However, there are additional techniques that can be used to further emphasize and refine the data to produce more accurate and timely results. Variable/feature weighting is used to emphasis certain sensor measurements over other variable/features and is often used in the feature selection process. In (Wang T. , 2013), an Empirical Signal-to-Noise Ratio (eSNR) is used for variable relevance evaluation. The eSNR is defined as

$$eSNR(s_i) = \frac{var(\underline{s_i})}{var(s_i)} \quad (2)$$

where s_i is a one dimensional time series representing the features of the system evolving over time. Let $\underline{s_i}$ be a smoothed version of s_i filtered by a certain filtering or smoothing algorithm. The idea is that, in the event the global variance (variance of the entire time series) is highly correlated to the local variance (variance within a shorter period of the time series), the smoothed time series will have a much smaller variance compared to the original. Therefore, the feature selection or emphasis can be performed from the ranking of the eSNR. The feature weighting is

$$\tilde{y}_n = y_n \cdot eSNR(y_n) \quad (3)$$

where n represents the n^{th} feature. This approach effectively de-emphasizes the features with large local variance.

Once the feature has been weighted, the next step is to uncorrelated the features. In this case, (Wang T. , 2013) suggests the use of Principal Component Analysis (PCA). PCA is a common technique used to transform the features into a smaller set of uncorrelated features. The uncorrelated feature will contain minimum redundancy and is important to combat the so-called curse of dimensionality. The method transforms the data into another coordinate system where the first coordinate or principal component (PC) represents the direction of the greatest variance of the original data with the second, third, etc. PC represents decreasing variance of the original data. The transformed features are calculated as

$$z = V_M^T \cdot (\tilde{y} - \overline{\tilde{y}}) \quad (4)$$

Where $\overline{\tilde{y}}$ is the mean of $\tilde{y}$, and V_M consist of the eigenvectors from the covariance matrix of $\tilde{y}$. The top M principal components that make up 90% of the total variance are retained. The resultant PCs form a new time series z for each training and testing instance. An example of variable weighting and dimensionality reduction of the original data can be seen in Figure 5. With the PCA completed, the original data is now ready for Degradation Trajectory Abstraction. The data is Figure 5 show how the system is degrading through time with the red line showing the degradation trajectory abstraction model discussed in the following section.

3. DEGRADATION MODELING/REGRESSION

The degradation models are built from the M Principal Components (PC) extracted from the normalized data as described in Section 2. These models describe the PCs of z as a function of time t:

$$G: z = {}^{l}g(t) + \varepsilon, 0 \le t \le {}^{l}t_I \quad (5)$$

where ε is the noise term and in many cases is modeled as Gaussian. (Wang, 2010)

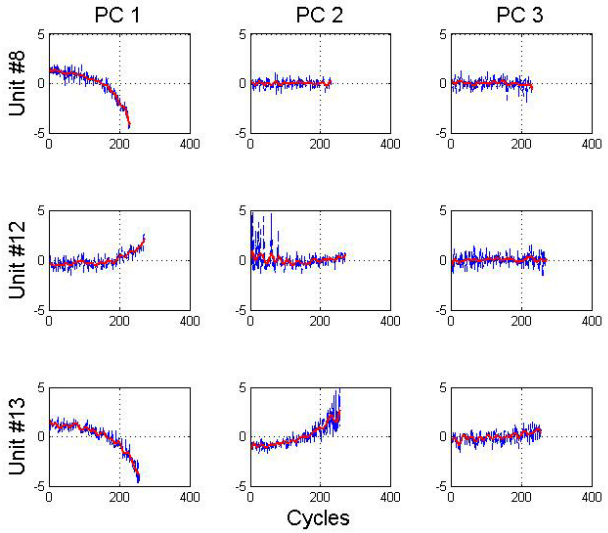


Figure 5. Example Trajectory Abstraction model from the "Turbofan Engine Degradation simulation" data set. The blue line is the variable weighting and dimensionality reduction of the original data, z. The red line is the degradation trajectory abstraction models.

There are many parametric and non-parametric methods that can be used to build the degradation models, all of which should be considered based on their ability to address the global degradation pattern, short-period characteristics, amount of available data, data noise level, and many other influential system characteristics. For this type of RUL estimation, long-term degradation behavior and the operating setting of the system are important, whereas the local fluctuations in the degradation trajectory can largely be considered noise. For these types of applications a smoothing operation of the time series such as a linear interpolation can be used. In (Wang, 2010), an exponential curve fitting, moving average filter and interpolation, Kernel regression smoothing, and relevance vector machines were explored.

Based on the results found in (Wang, 2010) the kernel regression smoothing approach was used for degradation Trajectory Abstraction in this paper; see Eq. (6)-(7).

$$z(t) = \frac{\sum_{i=1}^{E} K_G(t, t_i) \cdot z_i}{\sum_{i=1}^{E} K_G(t, t_i)} \quad (6)$$

$$K_G(x, y) = exp\left(\frac{\|x - y\|^2}{2\rho^2}\right) \quad (7)$$

Where ρ is the kernel width and is a free parameter usually chosen based on the data. An example output is shown in Figure 5 as the red line.

4. REMAINING USEFUL LIFE ESTIMATION

Once all the models have been trained, the testing data will need to be compared to every model and a similarity measure computed. The similarity measure is used to determine which model the system under test is most similar too. This can be done by computing a distance or similarity measure. In (Wang T. , 2013), the Minimum Euclidean Distance with Degradation Acceleration (MED-DA), Minimum Euclidean Distance with Time Lag (MED-TL), and Minimum Euclidean Distance with Time Lag and Degradation Acceleration (MED-TL-DA) was proposed. It was found that the MED-DA performed the best on the CMAPSS dataset evaluated. The remaining of this section, we briefly review MED-DA distance measure and provide an overview of two new similarity/distance measures we propose in this paper: Pearson's Correlation and Dynamic Time Warping.

4.1. Minimum Euclidean Distance with Degradation Acceleration

In (Wang T. , 2013), the Minimum Euclidean Distance with Degradation Acceleration (MED-DA) is the same as computing the Minimum Euclidean Distance between the training and testing models except the MED-DA uses a scaling factor for time dilation. This scaling factor is to accommodate the degradation rate differences between testing and training systems.

$${}^{l}D^2(\lambda) \coloneqq \frac{max(\lambda, \frac{1}{\lambda})}{I} \sum_{i=1}^{I} \sum_{m=1}^{M} \frac{(z_{mi} - {}^{l}g(\lambda \cdot t_i))^2}{2\sigma_m^2} \quad (8)$$

where $\max(\lambda, 1/\lambda)$ is the pentalty term for the difference in degradation rate.

The RUL prediction using this distance measure is calculated as:

$${}^{l}r_I = \frac{{}^{l}t_E}{arg\,\min_{\lambda}\, D^2(\lambda)} - t_I \quad (9)$$

Additionally, in (Wang T. , 2013) it was assumed that the most recent cycles provided more value to the similarity measure than the earlier cycles. Therefore (Wang T. , 2013) used a non-uniform weighting scheme to emphasis the most recent cycles of the system under test. Eq. (8) then becomes

$${}^{l}D_{DA}{}^{2}(\lambda) \coloneqq \frac{max(\lambda, \frac{1}{\lambda})}{I \sum_{i=1}^{I} \upsilon_i} \sum_{i=1}^{I} \upsilon_i \left(\sum_{m=1}^{M} \frac{(z_{mi} - {}^{l}g(\lambda \cdot t_i))^2}{2\sigma_m^2} \right) \quad (10)$$

where υ_i is the non-uniform weighting of each cycle i.

$$\upsilon_i = exp\left(-\frac{(t_i - t)^2}{2\rho^2}\right)$$
$$\rho = \gamma \cdot {}^{l}r_E \quad (11)$$

The non-uniform weighting is controlled by the spread parameter which is a percentage of the life ${}^{l}r_E$ of the degradation model ${}^{l}G$ and is controlled by the spread ratioγ. In (Wang T. , 2013), through cross-valuation, a spread parameter of 0.3 was found to produce the best results.

Since MED-DA is a squared distance measure, a similarity measure is computed as follows:

$${}^{l}S^{DA} = exp(-{}^{l}D_{DA}{}^{2}) \quad (12)$$

In (Wang T. , 2013), the best reported performance score on the evaluation set was 0.7534. This score is based the optimum values for the kernel width parameter ρ used for the kernel regression smoothing and spread ratio γ used in the MED-DA similarity evaluation. The optimum parameters were found by a 5-fold cross-validation of the training set where $\rho = 7$ and $\gamma = 0.3$.

4.2. Similarity based on Pearson's correlation

In (Lei & Govindaraju, 2004), a simple linear regression was used to assess the strength of a linear relationship between sequences $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$. A goodness-of-fit measures call $R^2$was used and is defined as:

$$R^2 = \frac{\sum_{i=1}^{n} u_i^2}{\sum_{i}^{n} (y_i - \overline{Y})} \quad (13)$$

where u is the error term and $\overline{Y}$ is the mean of Y. R^2 is also called the coefficient of determination. It is interpreted as the fraction of the variation in Y that is explained by X. After further evaluation it is found that R^2 is exactly the square of Pearson's correlation (Lei & Govindaraju, 2004).

$$S^{SLR} = r = \frac{\sum_{i}^{n} (x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum_{i}^{n} (x_i - \overline{X})^2} \sqrt{\sum_{i}^{n} (y_i - \overline{Y})^2}} \quad (14)$$

As r approaches 1, the linear relation between the two sequences becomes stronger. Therefore the Pearson's correlation of *X* and *Y* will have similarity *r*.

The RUL prediction using this similarity measure is a direct calculation between the test system and the model with the highest Pearson's correlation.

$${}^{l}r_I = {}^{l}t_E - t_I \quad (15)$$

4.3. Similarity based on Dynamic Time Warping

Dynamic Time Warping (DTW) is an alternative approach to determine the distance between two time-series signals where the two temporal sequences may vary in time or speed. It attempts to match two time series by "stretching" and "contracting" subsequences of the series so the difference between the series is minimized. (Giusti, 2013) The distance is then measured as the square root of the sum of the differences between the matched observations.

Technically, DTW (Salvador & Chan, 2007) constructs a warp path between the two time series. A dynamic programming approach is first used to find the warp path and create a cost matrix. A single point in the original time series can be warped to multiple points in the comparing time series. Every cell of the cost matrix is filled and the minimum-distance warp path can be evaluated by reversely following the smallest cost of each move until the original point is reached. If both series were identical, the warp path through the matrix would along the diagonal.

DTW can also adapt a constrained version by incorporating a window size parameter. This parameter limits the number of observations a matching can occur ahead or behind any given observation. It is noted in (Giusti, 2013) that the constrained version may sometimes improve the classification accuracy by avoiding pathological warping.

The RUL prediction using this similarity measure is a direct calculation between the test system and the model with the highest Pearson's correlation.

$$^{l}r_I = {}^{l}t_E - t_I \tag{16}$$

Since DTW is a squared distance measure, a similarity measure is computed as follows:

$$^{l}S^{DTW} = exp(-{}^{l}D_{DTW}{}^{2}) \tag{17}$$

4.4. Model Aggregation

All RUL estimates and similarity scores are used to form a hypothesis set and the goal of model aggregation is to use multiple estimates in the hypothesis set and sum them up to create a final prediction. The simplest method of aggregation is to use the similarity-weighted sum, which provides a Point Estimate of the RUL.

$$r_I := \frac{\sum_{l=1}^{L} {}^{l}S_I \cdot {}^{l}r_I}{\sum_{l=1}^{L} {}^{l}S_I} \tag{18}$$

This approach is inadequate for uncertainty management in prognostics. A probability distribution or confidence interval for the predicted RUL is desired in order to aid risk-informed decision-making in the context of prognostics and health management. (Wang, 2010)

5. UNCERTAINTY QUANTIFICATION IN RUL PREDICTION

The computation of uncertainty in the remaining useful life prediction is an important, essential, and challenging issue. Since prognostics deals with the prediction of the future behavior of engineering systems, it is necessary to understand that it is almost impossible to make predictions regarding the future. That is why it is important to quantify the various sources of uncertainty in prognostics and quantify their combined effect on the remaining useful life prediction.

Some recent research efforts in (Sankararaman, Daigle, & Goebel, 2014) and (Sankararaman & Goebel, 2013) have been focusing on the topic of quantifying the uncertainty in prognostics and the remaining useful life prediction. At any given instant of time at which prediction needs to be performed, the uncertainty in the RUL prediction depends on three important factors:

- Health state estimate at the time of prediction (initial state)
- Future operating and loading conditions
- Degradation model that predicts health state degradation from the initial state, based on the future operating and loading conditions

It has been demonstrated that the computation of the uncertainty in the RUL, based on the uncertainty in the above quantities is a non-trivial problem and needs to be solved using statistical methods (Sankararaman, 2014). In this context, the goal is to calculate the probability distribution of the remaining useful life prediction continuously as a function of time; note that this probability distribution varies as a function of time and therefore, needs to be recalculated at every time instant. This probability distribution needs to systematically account for the different sources of uncertainty in the aforementioned list of quantities and quantify their combined effect on prognostics and remaining useful life prediction.

Most of the previous efforts have focused on such uncertainty quantification only in the context of model-based prognostics where physics-based models are used to represent health state degradation. Uncertainty quantification and management in the context of data-driven prognostics has not been studied in the detail, and since, different types of data-driven techniques have been used by several researchers, the interpretation, quantification, and management of uncertainty may be different for different data-driven approaches. Hence, uncertainty quantification needs to be discussed in the context of the data-driven approach being pursued, and hence, this paper focuses only on uncertainty quantification in the TBSP approach.

5.1. Uncertainty in Similarity-Based Prediction Technique

In the context of similarity-based prediction, it is first essential to understand the importance of uncertainty quantification. In this methodology, the focus is on finding out the similarity between the desired testing data set and the entire training data set. The remaining useful life of the testing data set can be predicted through some sort of meaningful "interpolation" in the domain of the training data set, where the interpolation procedure attempts to identify where the testing data set lies, with respect to the training data set. An important underlying assumption here is that, at any point of prediction, the future operating conditions and loading conditions in the testing data set can also be interpolated based on that of the training data set; in many practical applications, this assumption may be incorrect and therefore, this method may not be applicable.

Therefore, if there is exact similarity between a testing data set and a particular training data set, then there is no uncertainty regarding the prediction of remaining useful life. This is because the remaining useful life of the desired testing data set is equal to the remaining useful life of the corresponding training data set. This can be easily explained by understanding data-driven learning algorithms such as Gaussian process learning where the variance of the prediction at any training point is exactly equal to zero. Therefore, if the testing point is identical to a training point, the variance of the prediction is zero and hence, there is no uncertainty regarding the remaining useful life. (Note that, the similarity-based comparison is performed only until the time of prediction. There may be significant differences between the testing set and the training set after the time of prediction; such differences lead to uncertainty in the remaining useful life prediction but cannot be quantified without knowledge regarding the future operating/loading conditions of the testing data set.)

Typically, the testing data set may be significantly different from the training data set, and the TBSP approach computes a similarity between the training and testing data set. This similarity measure is simply reflective of the probabilistic weightage that is given to each of the remaining useful life values of the training data set. Therefore, Eq. (18) implies that the remaining useful life is calculated only using a weighted averaging approach, and therefore, is reflective only of the mean behavior. Other statistics of the remaining useful life prediction can also be calculated. For example, the standard deviation can be calculated as:

$$\sigma_r = \sqrt{\frac{\sum_{l=1}^{L'} {}^{l}S_I ({}^{l}r_I - r_I)^2}{\sum_{l=1}^{L'} {}^{l}S_I}} \sqrt{\frac{L'}{L'-1}} \qquad (19)$$

where L' denotes the number of non-zero similarity measures.

Note that the weighted mean and weighted standard deviation are central measures. While such central measures are important, they do not sufficiently capture the information regarding the uncertainty in the remaining useful life prediction. In order to achieve this goal, it is necessary to calculate the entire probability distribution (either in terms of the probability density function or in terms of the cumulative distribution function). This calculation is facilitated through the use of kernel density estimation, as explained later in this section.

5.2. Uncertainty Quantification through Maximum Likelihood Estimation

In (Fonseca, Friswell, Mottershead, Lees, & Adhikari, 2005), the authors describe that the key to the maximum likelihood (ML) approach is to parameterize the probability density functions (PDFs) of the parameters. The uncertainty quantification includes calculating the probability that the measurements occur given the PDF of the parameters.

Suppose that the physical parameters, x, follow a certain probability distribution belonging to a probability distribution family parameterized by Θ (for example the mean, μ, and covariance matrix, Σ). For a given Θ, the output PDF, $f(x|\Theta)$, can be approximated using the uncertainty propagation method. Let the measurements be $x_1, x_2, \ldots, x_N$. The measurements are assumed to be independent, therefore the measurements likelihood is

$$L(\Theta) = f(x_1, x_2, \ldots, x_N|\Theta) = \prod_{i=1}^{N} f(x_i|\Theta) \qquad (20)$$

The maximum likelihood estimator is value of Θ that corresponds to the maximum of $L(\Theta)$. Note that the maximum likelihood estimate is also a central measure.

Two important changes need to be made in order to adapt this methodology for the purpose of uncertainty quantification in TBSP. First, it is necessary to infer information regarding the uncertainty; such uncertainty can be expressed either in terms of the PDF $f(x)$ or in terms of confidence intervals. Secondly, and more importantly, the PDF $f(x|\Theta)$ corresponds a parametric probability distribution (with parameters Θ), and such a distribution may not be available. So, it may be necessary to use non-parametric distribution and directly estimate the PDF $f(x)$ without employing the use parameters Θ. In this paper, both of these goals are accomplished through the use of a weighted kernel density function that is not only parametric but also can directly compute confidence intervals on the quantity of interest, x in this case.

5.3. Uncertainty Quantification through Kernel Density Estimation

A non-parametric approach for model aggregation is used which is called Kernel Density Estimation or KDE using a

Parzen window method. (Wang T. , 2013) The kernel density approximation is given by:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x - x_i}{h}\right) \quad (21)$$

where K is the Gaussian kernel function and h is the bandwidth for density estimation. The Gaussian kernel function is defined as:

$$K(u) = \frac{1}{2\sqrt{\pi}}exp\left(\frac{u^2}{2}\right) \quad (22)$$

In (Wang T. , 2013) and in this paper, the KDE method via diffusion with automatic bandwidth selection as proposed in (Botev, Grotowski, & Kroese, 2010) was used.

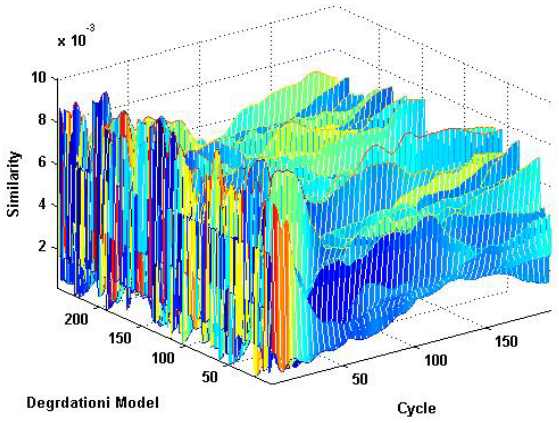


Figure 6. Example of SLR similarity between testing data and all degradation model for each cycle of the test system.

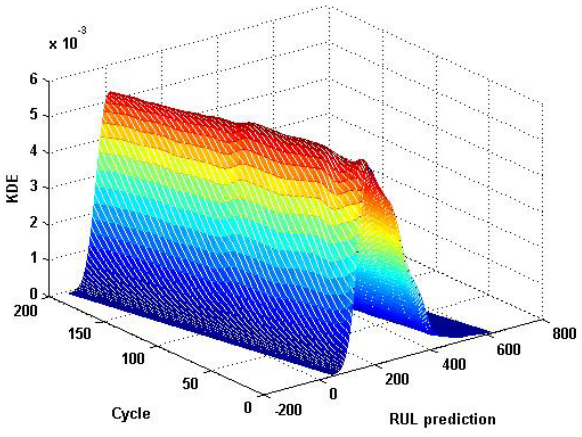


Figure 7. Kernel Density Estimation approach for RUL prediction using model aggregation.

Figure 6 shows an example of the similarity between test data and the trained models for over 200+ cycles. As can be seen in Figure 6, at the beginning the testing unit is very similar to all the degradation models, however as time (cycles) progresses the most similar degradation models can be readily observed. The plot in Figure 7 shows the density estimation of the RUL prediction at each cycle based on the SLR weighted KDE model aggregation.

6. PERFORMANCE METRICS

The evaluation of the proposed enhancements to TBSP will be based on the work in (Saxena, Celaya, Saha, Saha, & Goebel, 2009): Prediction Horizon, Rate of Acceptable Predictions, Relative Accuracy, and Convergence. A brief description of the metric will be provided in this section but the reader is referred to (Saxena, Celaya, Saha, Saha, & Goebel, 2009) and (Wang T. , 2013) for further information.

6.1. Prediction Horizon

Prediction Horizon (PH) is the time difference between the EoL failure and the time from which the RUL prediction first met the specified performance criteria,i.

$$PH = t_E - t_{i_a} \quad (23)$$

6.2. Rate of Acceptable Predictions

This metric quantifies the prediction quality. This is done by determining whether the prediction falls within a specified percentage of the true RUL for each RUL prediction.

$$AP = Mean(\{\delta_i | t_H \leq t_i \leq t_{EoUP}\}) \quad (24)$$

The specified percentage can be thought of as a cone of accuracy since as the true RUL decreases the accuracy requirement for the prediction become more stringent.

$$\delta_i = \begin{cases} 1 & if(1-\propto)r_i^* \leq r_i \leq (1+\propto)r_i^* \\ 0 & Otherwise \end{cases} \quad (25)$$

$$\delta_i = \begin{cases} 1 & \int_{r_i^*-\propto\cdot t_E}^{r_i^*+\propto\cdot t_E}\pi(r_i)dr_i \geq \beta \\ 0 & Otherwise \end{cases} \quad (26)$$

6.3. Relative Accuracy

Relative accuracy quantitatively evaluates the absolute percentage error of a prediction at a time within the prediction horizon, t_H , if the algorithm has met the requirements of the previous metrics.

$$RA = 1 - Mean\left(\left\{\frac{|r_i - r_i^*|}{r_i^*}|t_H \leq t_i \leq t_{EoUP}\right\}\right) \quad (27)$$

6.4. Convergence

Convergence evaluates how fast the prediction performance (any accuracy based metric) improves towards the end life of the instance, if the algorithm has met the requirements of the previous metrics.

$$CG = \left(\frac{\frac{1}{2}\sum_{i=P}^{E_oUP}(t_{i+1}^2 - t_i^2)M_i}{\sum_{i=P}^{E_oUP}(t_{i+1}^2 - t_i)M_i} - t_p \right) \cdot \frac{1}{t_{EoUP} - t_p} \quad (28)$$

6.5. Performance Score

The final evaluation metric or performance score used in (Wang T. , 2013) will be used in this paper. The performance score is a weighted sum of the Rate of Acceptable Predictions, Relative Accuracy, and Convergence.

$$PH = Median(\{^kPH\}) \quad (29)$$

$$AP = Median(\{^kAR\}) \quad (30)$$

$$RA = Median(\{^kRA\}) \quad (31)$$

$$CG = Median(\{^kCG\}) \quad (32)$$

Prediction Horizon is the only metric with a unit of time while the others have a value between 0 and 1, where 1 implies perfect. Since PH will be used as a preliminary requirement for the performance of RA, a weighted sum of the other three will be used as the overall performance score.

$$score = w_1 \cdot AP + w_2 \cdot RA + w_3 \cdot CG \quad (33)$$

where $w_1 = 0.6, w_2 = .3, w_3 = .1$. (Wang T. , 2013)

7. DATA SET& EXPERIMENT

To compare the performance of the proposed enhancements to the baseline TBSP in (Wang T. , 2013), this paper will use the same data set and experiment as outlined in (Wang T. , 2013).

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is used in this paper. C-MAPSS is a tool for simulating a realistic large commercial turbofan engine which simulates an engine model of a 90,000 lb thrust class turbofan engine that was written using MATLAB and Simulink. (Saxena A. , Goebel, Simon, & Eklund, 2008) There are four data sets of the run-to-failure data acquired from the C-MAPSS simulation (Saxena & Goebel, 2008). However, only the fourth data set, FD004, was used in (Wang T. , 2013) and will be used in this paper.

The data set FD004, has 2 fault modes, 6 operating condition regimes, 249 training units, and 248 testing units. There are 25 fields in the data set: cycle number, 3 condition settings, and 21 sensor measurements. Though FD004 provides a training and testing set, (Wang T. , 2013) determined that the testing set contained instances with incomplete run-to-failure data and would not be suitable for the performance evaluation method described in Section 6. Therefore, in (Wang T. , 2013) and in this paper the 249 training units are partitioned in to a training set of 150 randomly selected units with the remaining 99 units being used for evaluation.

For the experiment, the regime identification, mean variance normalization, and regression modeling follow the same procedure described in (Wang T. , 2013). For the RUL estimation, the SLR, MED-DA, and DTW are used to determine the similarity between the test system and the degradation models. The RUL of the test system is calculated based on four different approaches: 1) minimum distance (point estimation), 2) model aggregation (point estimation), 3) KDE (probability interval), and 4) MLE - Maximum Likelihood Estimation (confidence interval). In Summary, there are 12 different RUL predictions being evaluating for this paper; each similarity measure will have 2 point estimation, KDE, and a MLE.

8. RESULTS

This paper compares the RUL prediction using the similarity measure MED-DA from (Wang T. , 2013) to the Pearson's linear correlation coefficient and Dynamic Time Warping measures based on the FD004 data of the C-MAPSS data set.

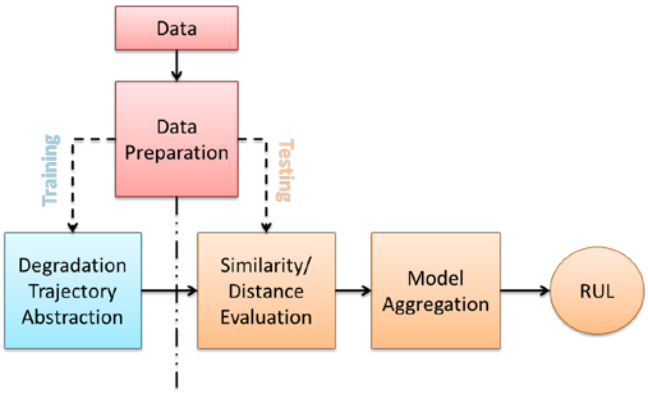


Figure 8. TSBP high level process flow (Wang T. , 2013)

The results are quite different than the ones report in (Wang T. , 2013). However, in (Wang T. , 2013) a single trial of 150 randomly selected units were used for training and the remaining 99 were used for testing. In this paper we performed our analysis using 20 independent trials. Figure 9 shows a boxplot of the 20 trial scores as defined in Eq. (33) showing the median performance of the 20 trials with the 25th and 75th percentiles as the edges of the box. The whiskers of the box extend to the most extreme data points not considered outliers, and the outliers are plotted individually.

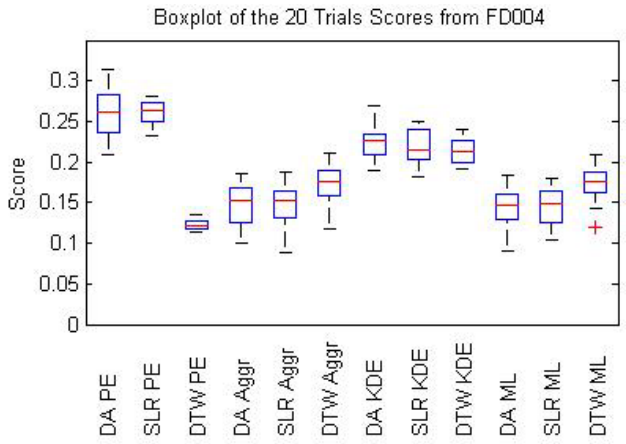


Figure 9. Boxplot of the 20 Trial Scores of 150 randomly selected units used for training and the remaining 99 used for testing.

The results in Figure 9 show that DA and SLR similarity measures are not significantly different for Point Estimation (PE), Aggregated (Aggr), Kernel Density Estimation (KDE), or Maximum Likelihood (ML) predictors. What is interesting is that the DTW measure performed worse than the DA and SLR measure using PE but outperformed them using a ML predictor. It is very difficult to form a conclusion based on the experiment performed by (Wang T. , 2013), because the results will be greatly dependent upon the randomly selected training and testing dataset. Hence, without knowledge of the specific randomly selected training model used for the results in (Wang T. , 2013), it is not feasible to perform analysis of all possible training model configuration to verify results.

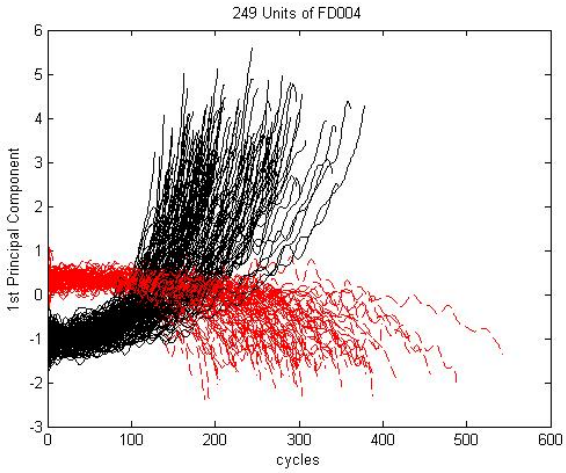


Figure 10. 249 Unit of the FD004 dataset. There were two fault modes identify in the dataset description file and can be clearly seen by the first principal component of the degradation model. Each line is 1st PC for the 249 units in the FD004 dataset.

9. BASELINE EXPERIMENT

Based on the above results we have decided to perform an additional experiment with the intent to baseline these measures and predictors for the FD004 dataset. In the baseline experiment we will make predictions for each of the 249 unit in the dataset. For each unit under test we will use the remaining 248 unit for training. This will allow the experiment to have maximum knowledge of the Fleet but without overlapping the degradation models and unit under test.

There are two fault modes identified in the FD004 dataset and can be seen in Figure 10. For simplicity, we will identify fault mode 1 as the red dashed lines and fault mode 2 as the solid black lines which have 101 and 148 degradation models, respectively. The only computational difference between the similarity evaluating of (Wang T. , 2013) and our baseline experiment is that we use only the degradation models for a given fault mode once it has been identified for the unit under test (UUT). The initial RUL predictions are based on all of the 248 degradation models, however after 30 or less cycles the UUT's fault mode is identified and the similarity comparisons is reduced to 101 or 148 degradation models. Of course at no time will the UUT degradation model be included in similarity computation of the training degradation models.

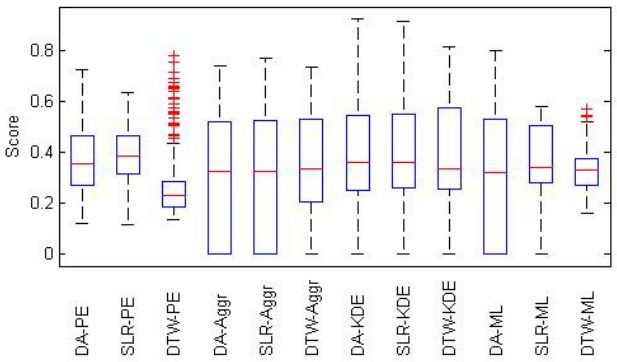


Figure 11. Boxplot of the baseline experiment.

Table 1. Median score performance of RUL similiarity-predictor combinations.

Pointe Estimate	DA-PE	SLR-PE	DTW-PE	Kernel Density Estimation	DA-KDE	SLR-KDE	DTW-KDE
20 Trials	0.2613	**0.2626**	0.1215	20 Trials	0.2267	0.214	0.2126
Baseline	0.2993	**0.3523**	0.1226	Baseline	0.2981	0.3035	0.2664
Model Aggregation	**DA-Aggr**	**SLR-Aggr**	**DTW-Aggr**	**Maximum Likelihood**	**DA-ML**	**SLR-ML**	**DTW-ML**
20 Trials	0.1528	0.1529	0.1769	20 Trials	0.1464	0.1492	0.1761
Baseline	0.2403	0.2327	0.2605	Baseline	0.2339	0.2808	0.2778

From Table 1, the SLR PE showed the best performance for both the 20 trial experiment adapted from (Wang T. , 2013)

and our established baseline experiment. However, these scores are based on point estimation performance metrics (Saxena, Celaya, Saha, Saha, & Goebel, 2009) and do not take advantage of the probability or confidence intervals of the ML or KDE predictors.

10. CONCLUSION

This paper examined alternative approaches to measure similarity in a Trajectory Based Similarity Prediction framework. Additionally, we evaluated a similarity weighted Kernel Density Estimation RUL predictor and similarity weight maximum likelihood RUL predictor. The use of these weighted KDE and ML predictors allows the RUL prediction to be defined over a probability and confidence interval. The two experiments presented show that the point estimation predictor using the Simple Linear Regression measure performed the best for each experiment, but further research will be needed to examine the benefit of the KDE and ML predictors that are not fully evaluated by the performance metrics

Some sources of error and uncertainty for TBSP approach include multi-regime normalization and sensor aggregation through principal component analysis, see Section 2. The regime normalization assumes uniform system degradation within and across the operational regimes which may greatly impact the similarity-predictor performance.

Additionally, it is very difficult and impractical to make predictions of RUL for systems that have an unknown operational profile. It is anticipate that real world systems will have a known operational profile with a desired maintenance free period for a given system. Therefore, predictions and prediction accuracies should be based on a failure occurring within a maintenance free period and known operational profile for certain applications. We envision future research will be focused with these restrictions in mind.

NOMENCLATURE

t_i	The time stamp of the i^{th} measurement cycle
z_i	The sample of PC vector at the i^{th} measurement cycle
E	The index of the End-of-Life measurement cycle for an instance.
P	The index of the Start-of-Prediction cycle for an instance.
$EoUP$	The index of End-of-Useful-Prediction cycle for an instance.
r_I	The estimated RUL at measurement cycle I
r_I^*	The ground-truth RUL at measurement cycle I.
^{l}	A left super script applied to any of the above symbols, indicating the symbol corresponding to the l^{th} training instance or degradation model.
^lG	The l^{th} degradation model extracted from the l^{th} training instance.
$^lD^2$	Squared distance to the l^{th} degradation model trajectory.
^lS	Similarity to the l^{th} degradation model trajectory.
$eSNR(\cdot)$	The Empirical Signal/Noise Ratio computed from 1-D time series data.
α	Percentage of RUL prediction error bound, e.g. 0.2.
i_α	The index of the first RUL prediction that satisfies the α-bound criteria.

REFERENCES

Botev, Z. I., Grotowski, J. F., & Kroese, D. P. (2010). Kernel density estimation via diffusion. *The Annals of Statistics*, 39(5), 2916-2957.

Dallachiesa, M., Nushi, B., Mirylenka, K., & Palpanas, T. (2012). *Uncertain Time-Series Similarity: Return to the Basics.* University of Trento.

Fonseca, J., Friswell, M. I., Mottershead, J. E., Lees, A. W., & Adhikari, S. (2005). Uncertainty Quantification using Maximum Likelihood: Experimental Validation. *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference.* Austin.

Giusti, G. R. (2013). An Empirical Comparison of Dissimilarity Measures for Time Series Classification. *Brazilian Conference on Intelligent Systems (BRACIS).* Fortaleza.

Gouriveau, R., Ramasso, E., & Zerhouni, N. (2013). Strategies to face imbalanced and unlabelled data in PHM applications. *Chemical Engineering Transactions*(33), 115-120.

Guo, H., Gerokostopoulos, A., Liao, H., & Niu, P. (2013). Modeling and Analysis for Degradation with an Initiation Time. *Reliability and Maintainability Symposium.*

Lei, H., & Govindaraju, V. (2004). Matching and Retrieving Sequential Patterns Under Regression. *IEEE/WIC/ACM International Conference on Web Intelligence.*

Otey, M. E., & Parthasarathy, S. (2004). A Dissimilarity Measure for Comparing Subsets of Data: Application to Multivariate Time Series. *Department of Computer Science and Engineering, The Ohio State University.*

Salvador, S., & Chan, P. (2007). FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis.*

Sankararaman, S. (2014). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, In press.

Sankararaman, S., & Goebel, K. (2013). Why is the Remaining Useful Life Prediction Uncertain? *Annual Conference of the Prognostics and Health Management Society 2013.*

Sankararaman, S., Daigle, M., & Goebel, K. (2014). Uncertainty Quantification in Remaining Useful Life Prediction Using First-Order Reliability Methods. *IEEE Transactions on Reliability*, 603-619.

Saxena, A., & Goebel, K. (2008). C-MAPSS Data Set. NASA Ames Prognostics Data Repository.

Saxena, A., & Goebel, K. (2008). *PHM08 Challenge Data Description*. Denver: 1st International Conference on Prognostics and Health Management.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., et al. (2008). Metrics for Evaluating Performance of Prognostic Techniques. *Prognostics and Health Management (PHM).*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostic performance metrics. *Prognostics and Health Management Society*.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. *Ist International Conference on Prognostics and Health Management (PHM08).* Denver.

Wang, P., & Coit, D. W. (2007). Reliability Assessment Based on Degradation Modeling with Random or Uncertain Failure Threshold. *Reliability and Maintainability Symposium.* Orlando, FL.

Wang, T. (2013). *Trajectory Based prediction for Remaining Useful Life Estimation.* Cincinnati: University of Cincinnati.

Yu, P., Yong, X., Datong, L., & Xiyuan, P. (2012). Sensor Selection with Grey Correlation Analysis for Remaining Useful Life Evaluation. *PHM Society Conference.*

Survey of Condition Indicators for Condition Monitoring Systems

Junda Zhu, Tom Nostrand, Cody Spiegel, and Brogan Morton

Renewable NRG Systems, *Hinesburg, Vermont, 05461, USA*
jz@renewablenrgsystems.com

ABSTRACT

Currently, the wind energy industry is swiftly changing its maintenance strategy from schedule based maintenance to predictive based maintenance. Condition monitoring systems (CMS) play an important role in the predictive maintenance cycle. As condition monitoring systems are being adopted by more and more OEM and O&M service providers from the wind energy industry, it is crucial to effectively interpret the data generated by the CMS and initiate proactive processes to efficiently reduce the risk of potential component or system failure which often leads to down tower repair or gearbox replacement. The majority of CMS are designed and constructed based on vibration analysis which has been refined over the years by researchers and scientists. This paper provides detailed description and mathematical interpretation of a comprehensive selection of condition indicators for gears, bearings and shafts. Since different condition indicators are sensitive to different kind of failure modes, the application for each condition indicators were also discussed. The Time Synchronous Averaging (TSA) algorithm was applied as the signal processing method before the extraction of condition indicators for gears and shafts. Time Synchronous Resampling algorithm was applied to stabilize the shaft speed before the extraction of bearing condition indicators. Several case studies of real world wind turbine component failure detection using condition indicators were presented to demonstrate the effectiveness of certain condition indicators.

1. INTRODUCTION

As the global market of wind energy continuously grows over the recent years, the maintenance strategy of wind farms is evolving from schedule base maintenance to condition based maintenance. Scientists, researcher and engineers specialized in condition based monitoring techniques designed and utilized condition indicators to monitor and track the health status of the assets of interest. Condition indicators can be extracted from various signal sources including tradition vibration based signal from accelerometers, acoustic emission signal, oil condition signal and signal collected from SCADA systems. Different condition indicators were designed for different applications. Ideally, vibration based condition based monitoring techniques are very capable of detecting component fault signatures at high speed or intermediate sections of the wind turbine while acoustic emission based techniques are more capable of low speed or planetary section component fault detection.

Previously, Vecer et al (2005) summarized a comprehensive selection of condition indicators for gears along with some typical vibration signal analysis algorithms. Also, the National Renewable Energy Laboratory (NREL) published a document named 'Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis' in 2012 covered detailed information regarding lots of the common condition indicators. This paper summarized a great amount of the information from the above mentioned two reports. And the authors provided an industry perspective on how to utilize different CIs including those not only for gears but also for bearings and shafts on machine health status monitoring.

In general, the definition of condition indicators consists of two parts, the analysis algorithm and the statistical features. Analysis algorithm can be narrowband analysis, residual analysis and frequency/amplitude modulation analysis and so on. Statistical features include root mean square (RMS), kurtosis, crest factor, skewness, peak, peak to peak etc. A typical condition indicator can be expressed as narrowband kurtosis or residual RMS. Therefore, as a matter of fact, condition indicators are designed to describe the time or frequency domain signal waveform or analysis result from specific analysis algorithm in a statistical manner. Typical condition monitoring system data processing flowchart for gears is presented in Figure 1. In Figure 1, the incoming raw vibration signal were collected from the accelerometers and then goes into the Time Synchronous Averaging Algorithm (TSA) to remove noises that were not synchronous with the shaft rotating frequency. Time synchronous average signal

Junda Zhu et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

is calculated by dividing the vibration signal into one revolution sections (based on the once-per-revolution tachometer signal). Each single revolution section is resampled into a common length to eliminate variations in speed. Then all the equal length sections are combined and averaged. TSA is a vibration signal processing algorithm that calculates the average vibration caused by one revolution of the shaft under analysis. It converts the vibration from the time domain into the revolution (or order) domain and significantly reduces all vibration that is not synchronous with the shaft. Bechhoefer explained the algorithm and its derivation (Bechhoefer and Kingsley, 2009). The signal then goes through residual analysis algorithm. After that, statistical features are extracted from the residual analyzed vibration signal. Similarly, the raw vibration signal goes through narrowband analysis, energy operator analysis, Amplitude Modulation (AM) analysis and Frequency Modulation (FM) analysis. Accordingly, statistical features are extracted from the analyzed signals which are defined as condition indicators.

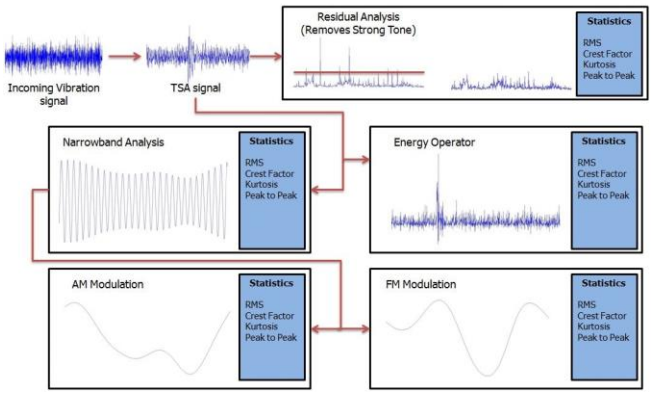


Figure 1. Vibration signal processing flow chart.

The first section of this paper gave an introduction to the techniques. The second section of this paper covered the definition of the statistical features, their definitions and applications. Then, the third section went over the analysis algorithms for different components including gears, bearing and shafts. The general descriptions of the analysis algorithm along with their applications were discussed. After that, the 4th session covered several case studies of real world wind turbine component failure detection using condition indicators to demonstrate the effectiveness of some of the described condition indicators. The last section summarized this paper.

2. STATISTICAL FEATURES

In general, statistical features were designed to describe the result of a specific vibration signal analysis algorithm. Common statistical features include Root Mean Square (RMS), Delta RMS, Peak, Peak to Peak, Kurtosis, Crest Factor, and Skewness, which were shown in the following respectively.

2.1. Root Mean Square (RMS)

RMS describes the energy content of the signal. RMS is used to evaluate the overall condition of the components. Therefore, it is not very sensitive to incipient fault but used to track general fault progression (Vecer et al, 2005).

$$s_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i)^2} \tag{1}$$

s_{rms} is the root mean square value of dataset *s*

s_i is the *i*-th member of points in dataset *s*.

N is the number of data points in dataset *s*.

2.2. Delta RMS

Delta RMS is the difference between two consequent RMS values.

$$Delta\ RMS = RMS(T) - RMS(T-1) \tag{2}$$

If the gear damage occurs, the vibration level will be increased more rapidly than in a normal case without gear damage (Vecer et al, 2005).

2.3. Peak

Peak value is the maximum amplitude of the signals within a certain time interval.

$$s_{peak} = max(s_1, s_2, s_3 \dots s_N) \tag{3}$$

Peak value is usually not used very often compared to peak to peak value.

2.4. Peak to Peak

Peak to peak value is the distance between the maximum amplitude and the minimum amplitude of the signal. Peak to peak is a measurement of spread in the signal.

$$s_{peak-peak} = s_{max} - s_{min} \tag{4}$$

2.5. Kurtosis

The shape of the amplitude distribution is often used as a data descriptor. Kurtosis describes how peaked or flat the distribution is. A kurtosis value close to 3 indicates a Gaussian-like signal. Signals with relatively sharp peaks have kurtosis greater than 3. Signals with relatively flat peaks have kurtosis less than 3. The following equation calculates the kurtosis (Vecer et al, 2005).

$$kurtosis = \frac{N \cdot \sum_{i=1}^{N}(s_i - \bar{s})^4}{\{\sum_{i=1}^{N}(s_i - \bar{s})^2\}^2} \tag{5}$$

N is the number of points in the history of signal *s*

s_i is the *i*-th point in the time history of signal *s*

Kurtosis provides a measure of size of the tails of distribution and is used as an indicator of major peaks in a set of data. As a gear wears and breaks, this feature should signal an error due to the increased level of vibration.

2.6. Crest Factor

Crest factor is the ratio of the single side peak value of the input signal to the RMS level (Vecer et al, 2005).

$$CF = \frac{s_{peak}}{s_{rms}} \tag{6}$$

CF is the crest factor

$s_{peak-peak}$ is the single side peak of the signal

s_{rms} is the root mean square value of the vibration signal

This value is normally between 2 to 6. Crest factor value over 6 indicates possible machine failure. There are certain variations on the definition of crest factor. The numerator could be the single side peak value (maximum or minimum) or a mean of the maximum and minimum of the signal of interest. Crest factor can be used to indicate faults in an early stage. This feature is used to detect changes in the signal pattern due to impulsive vibration sources such as tooth breakage on a gear.

2.7. Skewness

Skewness indicates the symmetry of the probability density function (PDF) of the amplitude of a time series. A time series with an equal number of large and small amplitude values has a skewness of zero. The following equation calculates skewness (Vecer et al, 2005).

$$Skewness = \frac{N \cdot \sum_{i=1}^{N}(s_i - \bar{s})^3}{\left\{\sqrt{\sum_{i=1}^{N}(s_i - \bar{s})^2}\right\}^3} \tag{7}$$

N is the number of points in the history of signal *s*

s_i is the *i*-th point in the time history of signal *s*

A time series with many small values and few large values is positively skewed (right tail), and the skewness value is positive. A time series with many large values and few small values is negatively skewed (left tail), and the skewness value is negative.

3. ANALYSIS ALGORITHMS

Analysis algorithms were applied before the extraction of statistical features. These algorithms were developed to enhance the component fault signatures. The statistical features extracted from the result of the algorithm are called condition indicators. Different condition indicators were developed to detect various faults on different components. This section categorizes them into three categories including bearing, shaft and gear. The typical analysis algorithm for different components were listed and explained along with the extracted condition indicators.

3.1. Bearings

Time Synchronous Resampling algorithm was applied to stabilize the shaft speed before the extraction of bearing condition indicators. In the CMS industry, it is common to have a hard threshold over certain shaft speed that triggers the data collection. Combined with TSR, the shaft speed can be controlled to a maximum extend in terms of speed fluctuation. In general, bearing fault characteristic frequencies are used to diagnose and localize the bearing fault induced by pitting, spall, cracking and etc. The specific bearing fault characteristic frequency of different components can be obtained from the bearing kinematic information. There are 4 common condition indicators for bearings which are ball energy, cage energy, inner race energy and outer race energy, respectively. A window of observation is usually set around the fault frequency of the bearings. This is designed to ensure even if the shaft speed is somewhat inaccurate, the amplitude of the bearing fault frequency can still be captured.

3.1.1. Ball Energy

Ball energy represents the energy of the bearing vibration signal at/around the rolling element fault frequency.

$$Energy_{ball} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_b \pm N)^2} \tag{8}$$

f_b is the fault frequency of the rolling element

N is half of the window of observation

3.1.2. Cage Energy

Cage energy represents the energy of the bearing vibration signal at/around the cage precession frequency.

$$Energy_{cage} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_c \pm N)^2} \tag{9}$$

f_c is the fault frequency of the cage

N is half of the window of observation

3.1.3. Inner Race Energy

Inner race energy represents the energy of the bearing vibration signal at/around the inner race fault frequency.

$$Energy_{Inner\ race} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i \pm N)^2} \quad (10)$$

f_i is the fault frequency of the inner race

N is half of the window of observation

3.1.4. Outer Race Energy

Outer race energy represents the energy of the bearing vibration signal at/around the outer race fault frequency.

$$Energy_{outer\ race} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_o \pm N)^2} \quad (11)$$

f_o is the fault frequency of the outer race

N is half of the window of observation

3.2. Shafts

All the condition indicators mentioned in this section were extracted after the original signal was processed through TSA algorithm. Typical condition indicator for shafts includes shaft order 1, shaft order 2, shaft order 3 and so on. Shaft condition indictors are used to detect shaft faults including shaft imbalance, misalignment etc.

3.2.1. RPM

Number of shaft revolution per minute. RPM is measurement of shaft speed. The 1/rev derivative of the RPM is a measurement of rated change of RPM at the 1/rev frequency. This measurement is capable of rotor shaft imbalance indication.

3.2.2. Shaft Order 1 (SO1)

Shaft Order 1 represents the magnitude of the first harmonics of the shaft of interest in frequency domain. SO1 is an indicator of mass imbalance or a bent shaft.

3.2.3. Shaft Order 2 (SO2)

Shaft Order 2 represents the magnitude of the second harmonics of the shaft of interest in the frequency domain. SO2 is sensitive to coupling failures (misalignment) or bent shaft.

3.2.4. Shaft Order 3 (SO3)

Shaft Order 3 represents the magnitude of the third harmonics of the shaft of interest in the frequency domain. SO3 is sensitive to coupling failures. For the main rotor, SO3 is driven by combined effect of tower shadow and wind shear.

3.2.5. TSA RMS

The root mean square value of the TSA signal

3.2.6. TSA Peak to Peak

The peak to peak value of the TSA signal

3.2.7. Shaft Order 1 Phase Angle

Phase angle can be calculated as four-quadrant inverse tangent of the complex conjugate FFT transform of the raw vibration signal. The phase angle of the shaft order 1. SO1 Phase Angle is an indication of imbalance.

3.2.8. 1/Rev Derivative of RPM

Rated shaft RPM change per revolution.

3.3. Gears

Among the condition indicators used on different components, condition indicators for gears normally involves a specific signal processing algorithm and a statistical feature. This section shows the common signal processing algorithm for gears and the condition indicators extracted from the analysis result that are often used.

3.3.1. Residual Analysis

The residual signal for a gear can be calculated by removing the shaft harmonics and the gear mesh frequency and harmonics from the time synchronous average signal. But the residual analysis algorithm can vary depends on the information the researchers trying to acquire or remove. Residual Signal is effective for detecting gear scuffing, tooth pitting and tooth crack faults. Periodic faults like tooth breakage normally can have impact of 1 per rev show up in the TSA signal. The residual analysis allows fault impact signatures to become prominent in the time domain.

Combined with the above mentioned statistical features, common condition indicators extracted from residual analysis are residual RMS, residual peak to peak, residual kurtosis, and residual crest factor.

3.3.2. Energy Ratio

Energy ratio is the ratio between the energy of the difference signal and the energy of the original meshing component (Vecer et al, 2005).

$$ER = \frac{\sigma(d)}{\sigma(r)} \quad (12)$$

$\sigma(d)$ is the standard deviation of the difference signal

$\sigma(r)$ is the standard deviation of the original signal

Energy ratio is very good indicator for heavy wear, where more than one tooth on the gear is damaged. The energy ratio will trend towards 1 as a fault progresses.

3.3.3. Energy Operator

Energy operator is computed as the normalized kurtosis from the signal where each point is computed as the difference of two squared neighborhood points of the original signal (Vecer et al, 2005).

$$EO = \frac{N^2 \sum_{i=1}^{N} (\Delta x_i - \Delta \bar{x})^4}{\{\sum_{i=1}^{N} (\Delta x_i - \Delta \bar{x})^2\}^2} \quad (13)$$

$\Delta \bar{x}$ is the mean value of signal Δx

$\Delta x_i = s_{i+1}^2 - s_i^2$

N is the number of data point in the dataset *x*

Energy Operator is a type of residual of the autocorrelation function. It is designed to reveal the amplitude modulations and phase modulations of the signal of interest. For a nominal gear, the predominant vibration is gear mesh. Surface disturbances and scuffing generate small higher frequency values, which are not removed by autocorrelation. Large energy operator indicates server pitting or scuffing.

Combined with statistical features, common condition indicators extracted from energy operator analysis are EO RMS, EO peak to peak, EO kurtosis, and EO crest factor.

3.3.4. FM0

FM0 is defined as the peak to peak level of the TSA signal divided by the sum of the amplitude at the gear mesh frequency and its corresponding harmonics (Vecer et al, 2005; Lebold et al, 2000).

$$FM0 = \frac{s_{peak-peak}}{\sum_{i=1}^{N} A(i)} \quad (14)$$

FM0 is the zero-order figure of merit

$s_{peak-peak}$ is the peak to peak value of the TSA signal.

A(i) is the amplitude of the i^{th} mesh frequency harmonics

FM 0 is a statistic used to detect major changes in the meshing pattern. For heavy wear, the peak to peak value remains constant while the meshing frequency decreases, causing the FM0 parameter to increase. FM0 is a generalized gear fault indicator, sensitive to gear wear/scuffing/pitting and tooth bending due to crack root. However, FM0 is not a good indicator for minor tooth damage.

3.3.5. Sideband Modulation Lifting Factor (SMLF)

Sideband modulation lifting factor (SMLF) or sideband level factor (SLF) is defined as the sum of the first order side band about the fundamental gear mesh frequency divided by the standard deviation of the signal of interest (Vecer et al, 2005).

$$SMLF = \frac{\sum_{i=1}^{n} si_{gear\ mesh \pm i}}{s_{std}} \quad (15)$$

si is the amplitude of the i^{th} sideband around fundamental gear meshing frequency

s_{std} is the standard deviation of the time signal average.

This parameter is based on the idea that tooth damage will produce amplitude modulation of the vibration signal. This CI is designed to detect gear misalignment.

3.3.6. G2

G2 is defined as the amplitude of the 2nd harmonics of gear meshing frequency over the amplitude of the gear meshing frequency in the frequency domain.

3.3.7. Narrowband (NB) Analysis

Narrowband analysis operates the TSA signal (or other time domain signal of interest) by filtering out all the tones except that of the gear mesh and with a given bandwidth. Narrowband signal is calculated by zeroing the bins in the Fourier transform of the TSA except the gear mesh. Statistics features of the narrowband signal can be calculated to enhance the fault feature. Narrowband represents the vibration associate with the primary gear mesh frequency. Narrowband analysis can capture sideband modulation of the gear mesh due to misalignment, or detect a cracker/soft/broken tooth.

Combined with statistical features, common condition indicators extracted from narrowband analysis are NB RMS, NB peak to peak, NB kurtosis, and NB crest factor.

3.3.8. Amplitude Modulation (AM) Analysis

Amplitude Modulation (AM) analysis is the absolute value of the Hilbert transform of the narrowband signal (Bechhoefer, 2012), since primary gear meshing characteristics extracted from narrowband analysis is the subject of interest. However, AM analysis is not limited to narrowband signal.

Modulation is a non-linear effect in which several signals interact with one another to produce new signals with frequencies not present in the original signals. Amplitude modulation is defined as the multiplication of one time-domain signal by another time-domain signal. For a gear with minimum transmission error, the AM analysis feature should be a constant value of gear tooth displacement. Gear defects or faults can increase the kurtosis of the signal significantly. AM is sensitive to eccentric gears and broken or soft tooth faults.

Combined with statistical features, common condition indicators extracted from AM analysis are AM RMS, AM peak to peak, AM kurtosis, and AM crest factor.

3.3.9. DAM

DAM is defined as the derivative of the amplitude modulation (AM) signal. DAM is sensitive to both soft and broken gear tooth faults.

Combined with statistical features, common condition indicators extracted from DAM analysis are DAM RMS, DAM peak to peak, DAM kurtosis, and DAM crest factor.

3.3.10. Frequency Modulation (FM) Analysis

Frequency Modulation (FM) is the derivative of the angle of the Hilbert transform of narrowband signal (Bechhoefer, 2012), since primary gear meshing characteristics extracted from narrowband analysis is the subject of interest. However, FM analysis is not limited to narrowband signal.

Modulation is a non-linear effect in which several signals interact with one another to produce new signals with frequencies not present in the original signals. Frequency modulation (FM) is the varying in frequency of one signal by the influence of another signal, usually of lower frequency. The frequency being modulated is called the carrier. Frequency Modulation analysis is in radians. Frequency modulation (FM) analysis is a powerful tool capable of detecting changes of phase due to uneven tooth loading, characteristics of a number of fault types. For certain gear architectures, FM analysis is more sensitive to fault than either the narrowband or amplitude modulation analysis.

Combined with statistical features, common condition indicators extracted from FM analysis are FM RMS, FM peak to peak, FM kurtosis, and FM crest factor.

3.3.11. FM4

FM4 is a simple measure if the amplitude distribution of the difference signal is peaked or flat. The mathematical representation is shown below. NA4 is determined by dividing the fourth statistical moment of the residual signal by the current run time averaged variance of the residual signal, raised to the second power (Vecer et al, 2005; Lebold et al, 2000).

$$FM4 = \frac{N \times \sum_{i=1}^{N}(d_i - \bar{d})^4}{\left\{\sum_{i=1}^{N}(d_i - \bar{d})^2\right\}^2} \quad (16)$$

d_i is the *i*-th point of the differential signal in the time record

N is the total number of points in the time record

The parameter assumes that a gearbox in good condition has a difference signal with a Gaussian amplitude distribution (kurtosis of 3), whereas a gearbox with a major peak or a series of major peaks results in a less peaked amplitude distribution (kurtosis greater than 3). For single tooth defect fault progression, the data distribution becomes peaky and the kurtosis increases. For multiple teeth fault progression, the data distribution becomes flat and the kurtosis value decreases.

3.3.12. NB4

NB4 is designed from the NA4 parameter.NA4 is calculated from the residual signal while NB4 uses the envelop of a band-passed segment of the time synchronous averaged signal.NB4 is determined by dividing the 4th statistical moments of the envelop signal, raised to the 2nd power. (Lebold et al, 2000; Lebold et al, 2000).

$$NB4 = \frac{N \times \sum_{i=1}^{N}(E_i - \bar{E})^4}{\left\{\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N}(E_{ij} - \bar{E}_j)^2\right\}^2} \quad (17)$$

E is the envelop of the band passed signal

$\bar{E}$ is the mean value of the enveloped signal.

N is the total data points in time record.

M is the current time record in the run ensemble.

$$E = |\bar{s}(t)| = \sqrt{s^2(t) + \tilde{s}^2(t)} \quad (18)$$

$|\bar{s}(t)|$ is the envelope of the analytic signal

$s(t)$ is an input analog signal

$\tilde{s}(t)$Is the Hilbert transform of the input signal

A few damaged gear teeth will cause transient load fluctuations that are different from normal tooth load fluctuations. The theory suggests these fluctuations will be manifested in the envelop of a signal which is band-pass filtered about the dominant meshing frequency.

3.3.13. NA4

NA4 is determined by dividing the fourth statistical moment of the residual signal by the current run time averaged variance of the residual signal, raised to the second power (Vecer et al, 2005; Lebold et al, 2000).

$$NA4 = \frac{N \times \sum_{i=1}^{N}(r_i - \bar{r})^4}{\left\{\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N}(r_{ij} - \bar{r}_j)^2\right\}^2} \quad (19)$$

r_i is the i-th point in the time record of the residual signal.

r_{ij} is the i-th point in the j-th time record of the residual signal.

j is the current time record

i is the data point number per reading

M is the current time record in the run ensemble

N is the number of points in the time record

3.3.14. NA4*

NA4* is an enhanced version of NA4. The improvement is achieved by normalizing the fourth statistical moment with the residual signal variance for a gearbox in good condition instead of the running variance, which is used for NA4 (Vecer et al, 2005; Lebold et al, 2000).

$$NA4^* = \frac{N \times \sum_{i=1}^{N}(r_i - \bar{r})^4}{(var(r_{OK}))^2} \quad (20)$$

$var(r_{OK})$ is the variance of the residual signal for a gearbox in good condition (obtained from a well-functioning gearbox)

When gear damage progresses, the averaged variance value increases rapidly which results in the decrease of the $NA4^*$ parameter. To overcome this problem $NA4^*$ is developed to be more robust when progressive damage occurs.

4. CASE STUDIES

This section presents three case studies covering gear, bearing and shaft. All the case studies are from the wind energy industry where there is a pressing need for condition monitoring systems. For the next three case studies, all data was collected and processed by TurbinePhD system.

4.1. Wind Turbine High Speed Pinion

The purpose of installing a condition monitoring systems is to help mitigate the high financial risk of unplanned maintenance and establish the framework for a new predictive maintenance program. A well developed condition monitoring systems should be capable of monitoring every bearing, gear and shaft in the gearbox as well as the generator and main bearing.

A condition monitoring system is designed to detect faults early on so that wind farm operators have the longest possible time to plan a maintenance action. This early detection is critical in avoiding secondary damage from catastrophic failure and the subsequent additional financial cost. Additionally, the system uses numerous complex algorithms to track the condition of a component, which in turn are then normalized and combined to estimate the overall health of the component. The result is excellent fault discrimination, which is arguably one of the most important aspects of a condition monitoring system. Fault discrimination is the ability to separate out a faulted component from good components. If the fault discrimination is good, then the alarms the system provides are trustworthy and actionable. On the other hand, if the fault discrimination is poor, then the likelihood of false alarms and missed detections increases. Finally, the system uses a patented automated diagnostic capability to provide the user with an easy to read display of which turbines need attention all through a cloud-based client interface. Thus, eliminating the need for complex data processing and interpretation before a maintenance decision can be made.

After installation, the condition monitoring systems gathered wind turbine fleet vibration data for two weeks at which point alarm and warning thresholds were generated. These thresholds are data driven values obtained by statistically eliminating the outlying abnormal components on each turbine that define if a component is damaged. Once the thresholds were established, an alarm was triggered for the High Speed Pinion (the last gear in the gearbox before the generator) on one of the turbines. Alarms are triggered when one or more Condition Indicators or CIs were elevated over the generated thresholds. In this case, several CIs were elevated while others were not. Since different CIs are sensitive to different fault modes, the type of fault can be estimated solely based on which CIs are elevated and which are not. From the list of CIs that responded to this fault, there was strong evidence that the alarm was triggered by a broken tooth. The wind farm operators were notified and an up tower visual inspection revealed the cracked tooth.

One of the Condition Indicators that is very sensitive to gear tooth pitting, scuffing and bending is called the FM0. It compares the general vibration level with the amplitude of gear meshing. A high FM0 value indicates the general vibration level is higher than normal and the gear meshing characteristic frequency is submerged in the high noise floor. In this case, FM0 was elevated to the point where the fault discrimination was perfect, meaning there were absolutely no overlapping values between the FM0 tracking the broken pinion and the FM0 tracking normal pinions on other turbines as seen in the following Figure 2. This means the probability of a false alarm or missed detection was extremely low.

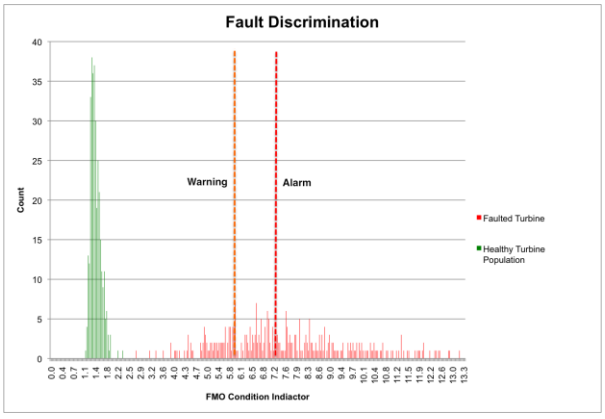


Figure 2. Fault discrimination based on FM0

While the FM0 Condition Indicator contributed to the triggered the alarm, other condition indicators were less sensitive to the fault. As explained previously, a condition monitoring system should offer clients the capability of

determining not only which component is not operating at a nominal condition but also performing diagnostics. This is critical information when it comes to cost savings, as different fault modes require different maintenance actions. In this case, the AM Kurtosis CI, which is a sensitive indicator of eccentric gears but less so at capturing tooth damage, remained at the nominal level as seen in the following Figure 3.

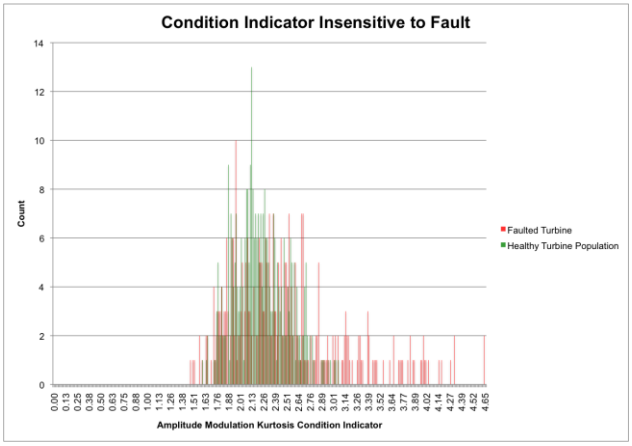


Figure 3. Fault discrimination based on AM Kurtosis

This specific turbine was shut down and inspected, the initial inspection found tooth damage on the high speed pinion as shown in the following Figure 4.

Figure 4. High speed pinion inspection result

Detecting this broken tooth early is critical for maintenance cost savings. When a gear loses a tooth, the remaining meshing teeth experience significant increases in load and subsequent stress and strain. This can cause cascading damage on the gear, which in turn will fill the gearbox with metal debris. Before long, other components are damaged and the gearbox potentially needs to be removed from the tower and rebuilt. A full gearbox rebuild, which requires the mobilization of a crane, can cost upwards of $150,000 and results in significant downtime, especially when climate can affect the ability to get a crane to the turbine. Additionally, a gear with a broken tooth, if left to run, will transfer damage to any gear that it is mated with. When this happens, both gears must be replaced. In this case, by implementing a well developed condition monitoring system, the wind farm operators obtained actionable information that left them with the option of performing an up-tower repair of just the High-Speed Pinion. The cost differential between performing this up-tower repair and a gearbox rebuild is estimated at $250,000. This proves that condition monitoring systems are valuable as a crucial part of the wind turbine maintenance cycle.

4.2. Wind Turbine High Speed Bearing

As mentioned earlier, the purpose of implementing a condition monitoring system is to help the wind farm operators to maximize the fleet availability by means of detecting the early damage of the drive train assembly before secondary damage occurs. Most retrofit condition monitoring systems need a certain period of time to gather data and thresholding, a process that defines the data characteristics of healthy components. Following the system thresholding, the Health Indicator (HI) of a "High Speed Bearing" (The bearing that holds the high speed generator shaft) started trending in March. The HI exceeded the warning and alarm limit around May.

The recommendation is when the HI exceeds the threshold of 1, an inspection should be performed on this component. The wind farm O&M team confirmed the bearing inner race fault and replaced the HS bearing. When the turbine started up and condition monitoring recommenced, the HI value dropped to below 0.2 indicating a nominal component.

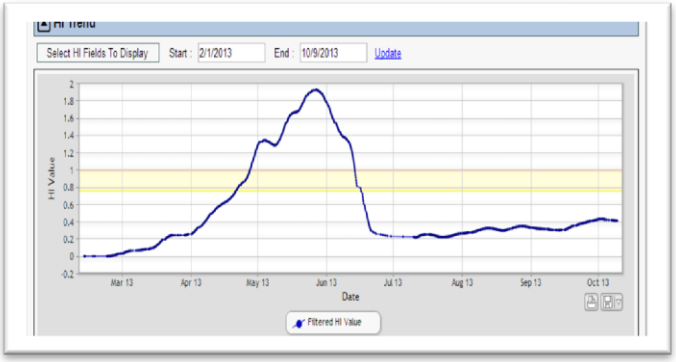


Figure 5. High speed bearing health indicator

The High speed bearing detail components CIs are also listed in the client interface as shown in Figure 6. From the pattern of the CI data log, the outer race, cage and rolling element energy showed no signs of degradation except the energy of the inner race. The inner race energy started increasing at March. Around May, the at the same time high speed bearing HI exceeds alarm limit, the inner race CI also exceed its own alarm threshold. This confirms that the HS bearing inner race cased the failure. The inner race fault had been located in March. The TurbinePHD systems tracked the fault progressing over a 2 month period. After HS

shaft/bearing replacement, the inner race energy dropped back to nominal.

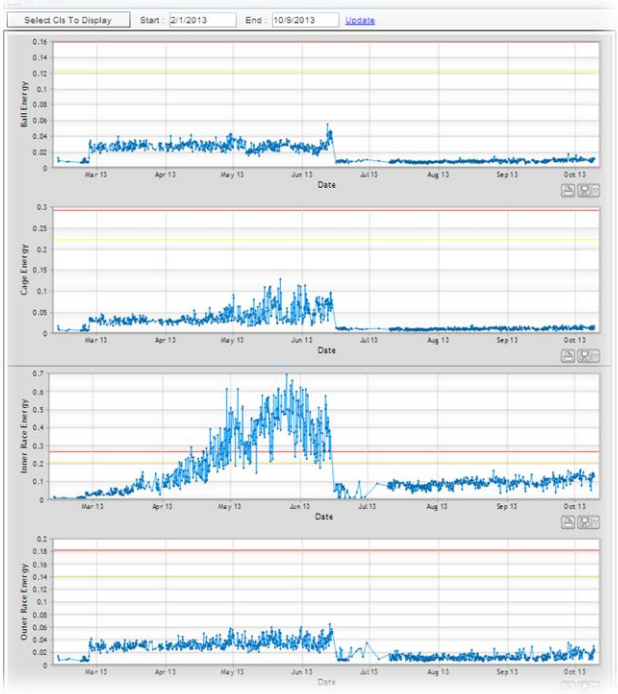


Figure 6. High speed bearing component trend

Based on the inner race details presented in Figure 3 and 4, which is available using the client web interface of TurbinePHD. One can observe that the condition indicator has pick up the inner race fault and starts trending 2 month before the condition indicator exceeded the alarm threshold. In the component detail page of the Web interface (Figure 4) the spectral information is displayed in the frequency domain. A high magnitude peak around the inner race fault frequency with characteristic sidebands that are a product of the shaft modulation can be seen.

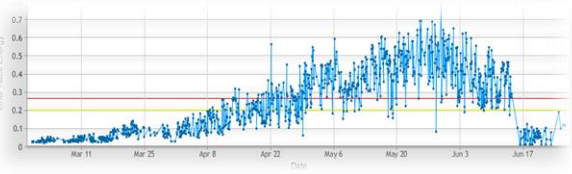


Figure 7. A detail look at the inner race condition indicator

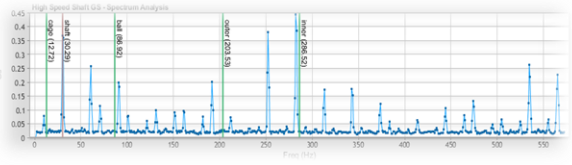


Figure 8. Spectrum analysis showed a high magnitude peak around the inner race fault frequency

After the O&M bore scope inspection, a large crack was found on the inner race which confirms the TurbinePHD diagnostics as shown in Figure 9.

Figure 9. Bore scope inspection of the inner race

4.3. Wind Turbine Rotor Imbalance

There can be many reasons behind a imbalanced rotor. In general, wind turbine rotor imbalance can be differentiating in the 2 types, Mass imbalance and aerodynamic imbalances. The imbalance can be induced by main reasons and some of them are listed as follow.

- Improper component manufacturing.
- Uneven buildup of debris on rotors, vanes or blades (ice, etc.).
- The addition of shaft fittings without an appropriate counter balancing procedure.
- Vane/blade erosion, crack or thrown balance weights. Fluid inclusion in the rotor blades.
- Rotor division error.
- Blade bearing jammed.
- Gearbox support structure excessive wear and tear.
- Generator alignment loss and coupler damage.
- Support structure and main frame damage.
- Yaw system/yaw breaks excessive wear and tear.
- Door frame damage, cracks at welds top and bottom, steps.
- Foundation bolt failure.

The effects of rotor imbalance include the following.

- 35% of all wind turbines have rotor caused vibrations which exceed the designed specifications. These vibrations cause unusual structure loads, an increased

wear, adverse startup conditions and often vibration causing emergency turn off.

- Rotational excitations cause higher dynamic load beyond design specification on bearing which leads to bearing failure from early fatigue. Fatigue, in a bearing, is the result of stresses applied immediately below the load carrying surfaces and is observed as appalling away of surface material.
- A wind turbine with an unbalanced rotor will lose some of its low wind production capability.
- High level of rotor vibration that appear as high magnitude of 1st harmonics of shaft rotating frequency.
- High levels of vibration caused by rotor imbalance results in turbine efficiency loss.

Rotor unbalance is a leading contributor to the need for frequent and costly maintenance action on yaw systems and fastening hardware. The unbalanced force on the rotor causes a reaction on the yaw system twice per revolution, accelerating the wear on the yaw gear teeth through impact loading and adding to the fatigue loading of the tower shell and mounting bolts.

A Leading wind energy operator asked Renewable NRG Systems to instrument their MW class turbine fleets with the TurbinePHD Condition Monitoring System to help them maximize the turbine availability by means of detecting the early damage of the drive train assembly before any secondary damage occurs. Following the standard commissioning procedure, the system ran for two weeks gathering data and was then thresholded, a process that establishes data driven definitions of when a component is no longer nominal. Following the system thresholding it was immediately apparent that "Nacelle X" (a component that watches the sway of the turbine tower) was not "nominal".

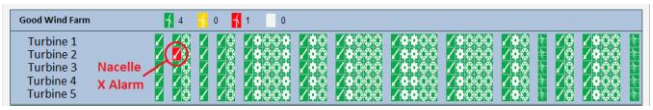


Figure 10. TurbinePHD Cloud Based Client Interface

A quick click on the red component revealed the Health Indicator (HI) value was elevated because the tower was swaying at the rotational frequency of the main rotor. This condition is a typical characteristic of a heavy blade and the subsequent imbalance (once per revolution imbalance). The recommendation is that when the HI exceeds the threshold of 1 an inspection needs to be performed on these component/components. In this case the HI value was floating around 1 between March 12th and June 13th. The wind farm O&M team inspected the blades and found that a heavy blade was causing the imbalance. The other turbine blades had a weight adjustment and subsequently the HI value dropped to nominal. After the 13th there was no data for a month because the turbine was down for maintenance. When the turbine started up and condition monitoring recommenced, the HI value had dropped to below .2 indicating a nominal component.

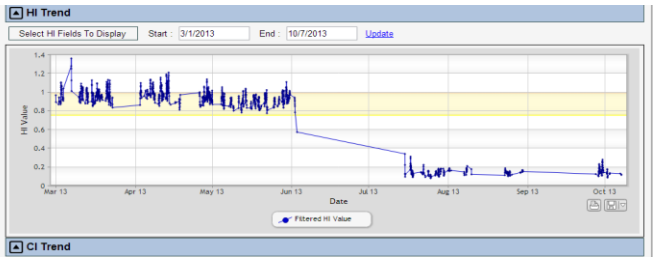


Figure 11. Health Indicator Trend

The Health Condition (HI) represents the data fusing result of all the Condition Indicators (CI). In TurbinePHD The shaft condition indicators includes shaft order 1 (SO1), shaft order 2 (SO2), shaft order 3 (SO3), 1 per revolution delta RPM and etc.

In this case, compared to SO2 and SO3, SO1 is trending along with the HI. The trending pattern correlates well between SO1 and component HI. The trending of SO1 confirmed the reason behind the high HI is because of the imbalance of the Rotor. Meanwhile, the CI on the Tach component, 1/rev dRPM, showed the same patter between March and October.

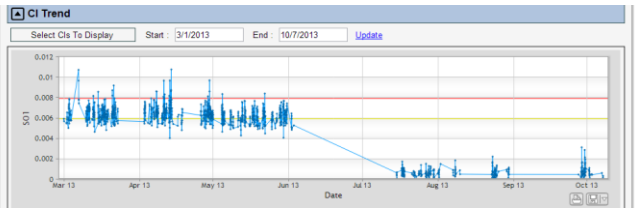


Figure 12. 1st shaft order (SO1), a measurement of the energy associated with the rotational frequency of the rotor. SO1 is one of several Condition Indicators (CIs) that are used to calculate the HI.

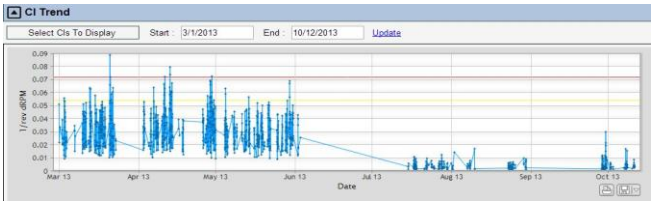


Figure 13. 1/rev dRPM, a measurement of rated change of RPM at the 1/rev frequency. 1/rev dRPM is one of the several Condition Indicators that are used to calculate Component HI of the Tach.

5. CONCLUSION

Condition indicators play a significant role in machine health status monitoring and tracking. Over the years, scientists and researchers have developed a great selection

of condition indicator for various components and applications. These condition indicators provides insights of the components condition and increase the signal storage and transmitting efficiency at the same time. Therefore, condition indicators are widely accepted by researchers and engineers for vibration signal analysis, acoustic emission signal analysis and sometimes oil debris and oil condition analysis as well.

This paper provided a detailed description and mathematical interpretation of a comprehensive selection of condition indicators developed for gears, bearings and shafts. Since different condition indicators are sensitive to different kind of failure modes, the application for each condition indicators were explained and discussed. The Time Synchronous Averaging (TSA) and Time Synchronous Resampling (TSR) algorithm was applied as the signal processing method before the extraction of condition indicators by the authors. Several case studies of real world wind turbine component failure detection using condition indicators were presented to demonstrate the effectiveness of certain condition indicators.

REFERENCES

Antoni, J., (2002), Differential Diagnosis of Gear and Bearing Faults. *Journal of Vibration and Acoustics*, Vol. 124, No. 2, 2002; pp. 165 - 171. http://dx.doi.org/10.1115/1.1456906

Antoni, J., Randall, R.B. (2006), The Spectral Kurtosis: Application to the Vibratory Surveillance and Diagnostics of Rotating Machines, *Mechanical Systems and Signal Processing*, Vol. 20, No. 2, 2006, pp. 308 - 331.

Barszcz, T. & Randall, R.B. (2009), Application of spectral Kurtosis for Detection a Tooth Crack in the Planetary Gear of a Wind Turbine, *Mechanical Systems and Signal Processing*, Vol. 23, pp. 1352 – 1365.

Bechhoefer, E., & Kingsley, M. (2009). A Review of Time Synchronous Average Algorithms, *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, San Diego, CA Sep. 27 – Oct. 1, 2009

Bechhoefer, E (2004), Method and Apparatus For Determining The Health Of A Component Using Condition Indicators, US Patent No. US6728658.

Bechhoefer, E., (2013), An Enhanced Time Synchronous Averaging for Rotating Equipment Analysis, *Proceedings for the joint conference: Machinery Failure Prevention Technology 2013 and International Instrumentation Symposium 2013*, May 13 – May 17, Cleveland, OH.

Bechhoefer E. & Mayhew E., (2006), Mechanical Diagnostics System Engineering in IMS HUMS, *Proceedings of the International IEEE Aerospace Conference*, pp. 1 - 8.

Bechhoefer E., (2012), Analysis Algorithms and Diagnostics Results from NRG Systems, Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis, *Technical Report, NREL/TP-5000-54530*, July 2012, contract no. DE-AC36-08GO28308

Bonnardot, F., El Badaoui, M., Randall, R.B., Daniere, J, and Guillet, F., 2005, Use Of The Acceleration Signal Of a Gearbox in Order To Perform Angular Resampling (With Limited Speed Fluctuation), *Mechanical Systems and Signal Processing*, Vol. 19, No. 4, pp. 766 – 785.

Braun S., (2011), The Synchronous (Time Domain) Average Revisited, *Mechanical Systems and Signal Processing*, Vol. 25, pp. 1087 - 1102.

Combet, F., & Gelman, L., (2010), Novel Adaptation of the Demodulation Technique for Gear Damage Detection to the Variable Amplitude of Mesh Harmonics, *Mechanical Systems and Signal Processing*, Vol. 25, pp. 839 - 845.

Combet, F., & Gelman, L., (2007), An automated methodology for performing time synchronous averaging of a gearbox signal without speed sensor, *Mechanical Systems and Signal Processing*, Vol. 21, issue 6, August 2007, pp. 2590 - 2606.

Crabtree C., Zappala D. & Tavner P., (2014), Survey of Commercially Available Condition Monitoring Systems for Wind Turbines, *Technical Report*, Durham University School of Engineering and Computing Sciences and the SUPERGEN Wind Energy Technologies Consortium.

Decker H., & Zakrajsek J., (1999), Comparison of Interpolation Methods as Applied to Time Synchronous Averaging, NASA/TM – 1999 – 209086, ARL – TR – 1960.

Dempsey, P., (2000), A Comparison of Vibration and Oil Debris Gear Damage Detection Methods Applied to Pitting Damage, *Proceedings of the 13th International Congress on Condition Monitoring and Diagnostic Engineering Management*, December 3 - 8, 2000, Houston, Texas. NASA/TM-2000-210371. Cleveland, OH: National Aeronautics and Space Administration (NASA), Glenn Research Center, 2000; 18 pp.

Dempsey P., Afjeh A., (2002), Integrating Oil Debris and Vibration Gear Damage Detection Technologies Using Fuzzy Logic, International 58th Annual Forum and Technology Display, Quebec (Canada), Junda 11 – 13, 2002.

Felten, D., 2003, Understanding bearing vibration frequencies, Mechanical Field Service Department, L&S Electric, Inc., Schofield, Wisconsin, pp. 1 – 3.

Germanischer Lloyd. (2007), Guidelines for the Certification of Condition Monitoring Systems for Wind Turbines, Hamburg, Germany, 2007.

Hochmann, D. & Sadok, M. (2004), Theory of Synchronous Averaging, *Proceedings of the 2004 IEEE Aerospace Conference*, March 6 - 13, 2004, Big Sky, Montana. Washington, DC: IEEE, 2004; pp. 3636 - 3653.

Jardine, A.K.S., Lin D., & Banjevic D., (2006), A Review on Mahinery Diagnostics and Prognostics Implementing Condition-based Maintenance, *Mechanical Systems and Signal Processing*, Vol. 20, pp. 1483 – 1510.

LaCava, W., van Dam, Jeroen., McNiff, B., Sheng, S., Wallen, R., McDade, M., Lambert, S., & Butterfield, S., (2011), Gearbox Reliability Collaborative Project Report: Findings from Phase 1 and Phase 2 Testing. *NREL/TP-5000-51885*. Golden, CO: National Renewable Energy Laboratory, June 2011.

Lebold, M., McClintic K., Campbell, R., Byington C., & Maynard K. (2000), Review of Vibration Analysis Methods For Gearbox Diagnostics and Prognostics, *Proceedings of the 54th meeting of the Society for Machine Failure Technology*, Virginia Beach, VA, May 1 – 4, 2000, pp. 623 – 634.

Mba, D. and Rao, R., (2006), Development Of Acoustic Emission Technology For Condition Monitoring And Diagnosis Of Rotating Machines; Bearings, Pumps, Gearboxes, Engines And Rotating Structures, *The Shock and Vibration Digest*, Vol. 38, No. 1, pp. 3 – 16.

McFadden P.D., (1986), Detecting Fatigue Cracks in Gears by Amplitude and Phase Modulation Of The Meshing Vibration, *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 108, pp. 165 - 170.

McFadden, P.D., (1987), A Revised Model For The Extraction Of Periodic Waveforms By Time Domain Averaging, *Mechanical Systems and Signal Processing*, Vol. 1, No. 1, pp. 83 – 95.

McFadden, P.D., (1991), A Technique For Calculating The Time Domain Averages Of The Vibration Of The Individual Planet Gears And The Sun Gear In An Epicyclic Gearbox, *Journal of Sound and Vibration*, Vol. 144, No. 1, pp. 163 – 172.

McFadden, P.D.; Smith, J.D., (1984), Vibration Monitoring of Rolling Element Bearings by the High-Frequency Resonance Technique - A Review, *Tribology International*, Vol. 17, No. 1, pp. 3 - 10.

McFadden, P., & Smith, J. (1985), A Signal Processing Technique for Detecting Local Defects in a Gear from a Signal Average of the Vibration., *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 199, No. 4, 1985; pp. 287 - 292.

McFadden, P. D. and Toozhy, M. M., (2000), Application Of Synchronous Averaging To Vibration Monitoring Of Rolling Element Bearings, *Mechanical Systems and Signal Processing*, Vol. 14, No. 6, pp. 891 – 906.

Randall, R.B., (2011), Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications, *Wiley Publication*, ISBN-13: 978-0470747858, ISBN-10: 0470747854

Randall, R.B., Antoni, J., (2011), Rolling Element Bearing Diagnostics-A Tutorial, *Mechanical Systems and Signal Processing*, Vol. 25, No. 2, 2011; pp. 485 - 520.

Sawalhi N., Randall R., & Forrester D., (2012), Techniques for Separation and Enhancement of Various Components in the Analysis of Wind Turbine Vibration Signals, Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis, *Technical Report, NREL/TP-5000-54530*, July 2012, contract no. DE-AC36-08GO28308.

Sharma S. & Mahto D., (2013), Condition Monitoring of Wind Turbines: A Review, International Journal of Scientific Engineering Research, Vol. 4, Issue 8, PP. 35 – 50, August, 2013, ISSN 2229 – 5518.

Sheldon J., Watson M., Mott G. & Lee H., (2012), Combining Novel Approaches with Proven Algorithms for Robust Wind Turbine Gearbox Fault Detection, Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis, *Technical Report, NREL/TP-5000-54530*, July 2012, contract no. DE-AC36-08GO28308

Sheng, S. (2012). Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis, *Technical Report, NREL/TP-5000-54530*, July 2012, contract no. DE-AC36-08GO28308

Sheng, S. (2011), Investigation of Various Condition Monitoring Techniques Based on a Damaged Wind Turbine Gearbox., *Proceedings of the 8th International Workshop on Structural Health Monitoring*, 13-15 September 2011, Stanford, CA. NREL/CP-5000-51753. Golden, CO: National Renewable Energy Laboratory, 2011.

Siegel D., Lee J., & Dempsey P., (2014), Investigation and Evaluation of Condition Indicators, Variable Selection, and Health Indication Method and Algorithms for Rotorcraft Gear Components, *Proceedings of the Machine Failure Prevention Technology Conference 2014*, Virginia Beach, VA, May 20 – 22.

Siegel D., Zhao W., Lapira E., AbuAli M., & Lee J., (2012), Review and Application of Methods and Algorithms in Wind Turbine Gearbox Fault Detection, Wind Turbine Gearbox Condition Monitoring Round Robin Study – Vibration Analysis, *Technical Report, NREL/TP-5000-54530*, July 2012, contract no. DE-AC36-08GO28308.

Spectra Quest Tech Note. (2006), Analyzing Gearbox Degradation Using Time-Frequency Signature Analysis, March, 2006.

Vecer, P., Kreidl, M., &Smid, R. (2005), Condition Indicators for Gearbox Condition Monitoring Systems. *ACTA Polytechnica*. Vol. 45, No. 6, pp. 35 – 43.

BIOGRAPHIES

Junda Zhu received his B.S. degree in Mechanical Engineering from Northeastern University, Shenyang,

China, and M.S. degree in Mechanical Engineering from The University of Illinois at Chicago in 2009, and Ph.D. degree in Industrial Engineering and Operational Research from The University of Illinois at Chicago in 2013. Dr. Junda Zhu is a Systems Engineer in the Turbine Health Monitoring Group in Renewable NRG Systems. His current research interests include rotational machinery health monitoring, diagnosis and prognosis with vibration or acoustic emission based signal processing techniques, lubrication oil condition monitoring and degradation simulation and analysis, physics/data driven based machine failure modeling.

Tom Nostrand received his BS in Engineering from the University of New Hampshire. He is currently an Engineering Manager at Renewable NRG Systems, Turbine Products Group. He has been working in the wind power industry for 6 years. Prior to this he spent 20 years in the aerospace electronics field working on many different commercial aircraft platforms and systems.

Cody Spiegel received his B.S. in Mechanical Engineering from The University of Vermont in 2013. He began a mechanical engineering internship at Renewable NRG Systems in 2011 and is now an Associate Mechanical Engineer working in the Turbine Health Monitoring Group. His current work includes condition monitoring diagnostics and support as well as R&D of new condition monitoring systems.

Brogan Morton received his B.S. and M.S. in Mechanical Engineering from The University of New Hampshire in 2000 and 2002 major in estimation and controls. He received MBA from the Idaho State University major in technology development and deployment. Brogan is currently the product manager for the TurbinePHD condition monitoring system in Renewable NRG Systems. He is responsible for the full product life cycle of several technologically advanced products in the wind energy industry.

Estimation of Bogie Performance Criteria Through On-Board Condition Monitoring

Parham Shahidi[1], Dan Maraini[1], Brad Hopkins[1], and Andrew Seidel[1]

[1]*Amsted Rail Company, Inc. Chicago, IL, 60606, USA*

pshahidi@amstedrail.com
dmaraini@amstedrail.com
bhopkins@amstedrail.com
aseidel@amstedrail.com

ABSTRACT

In this paper, bogie performance criteria are reviewed and it is shown that a real-time, on-board condition monitoring system can efficiently monitor these criteria to improve failure mode detection in freight rail operations. Although the dynamics of rail car bogie performance are well understood in the industry, this topic has recently received renewed attention through impending regulatory changes. These changes seek to extend empty rail car performance criteria to include loaded rail cars as well. Currently, the monitoring of bogie performance is primarily accomplished by wayside detection systems in North America. These systems are only sparsely deployed in the track network and do not offer the ability to monitor bogies continuously. The lack of these elements leads to unexpected downtimes resulting in costly reactive maintenance and lengthy periods of time before an adequate performance history can be established. This paper reviews performance criteria which critically influence bogie performance and proposes a vibration based condition monitoring strategy to estimate system component deterioration and their contribution to the development of bogie hunting. The strategy addresses both sensing techniques and monitoring algorithms to maximize the efficiency of the monitoring solution. In particular it is proposed that understanding the relation of different hunting modes to car body oscillations can be used for a deeper understanding of the rail car condition which current technologies are not able to provide.

1. INTRODUCTION

A freight rail bogie is the main vehicle connecting the freight rail car body to the rail. Typical freight rail cars utilize two bogies underneath the car body to carry the lading. Railroad terminology refers to the most widely distributed bogie type in North America as the three-piece bogie. Figure 1 gives a general overview of the components of the three-piece bogie. The three main components of this system are the two side frames and connecting bolster.

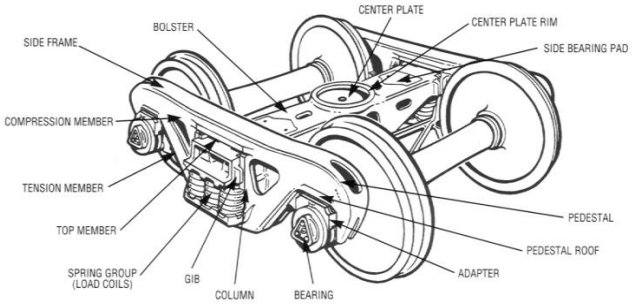


Figure 1. Standard North American three-piece bogie

This bogie type is also commonly used in Russia, China, Australia and most African countries. The bolster is connected to the side frames through a spring nest in each side frame which is referred to as the secondary or also central suspension. The two wheelsets are connected to the side frames by tapered roller bearings which are designed to maintain extremely high vertical and lateral loads. Many different sizes exist in North America carrying loads ranging from 177,000 to 315,000 lbs gross rail load (GRL). The bogie connects to the car body through the center plate.

The Association of American Railroads (AAR) is the standard setting organization for North America's railroads, focused on improving the safety and productivity of rail transportation. The AAR devises new rules for all aspects of rail transport, including freight car and bogie designs. Two major specifications exist, according to which all bogie systems intended for North American interchange service have to be designed. The first one is M-965, which was

Parham Shahidi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

adopted in 1968 and allowed for gross rail loads of up to 263,000 lbs. This rule was expanded in 2003 with the release of rule M-976 which was intended to regulate gross rail loads higher than 268,000 and up to 286,000 lbs. M-976 was directly related to AAR rule S-286 which sets the framework for the entire 286,000 GRL freight car. An extensive suite of tests exists which both M-965 and M-976 bogies have to pass in order to be approved for North American interchange service. This set of tests is formalized in the Manual of Standards and Recommended Practices (MSRP) C-II Chapter 11 (AAR, 2007) which contains the trackworthiness criteria limits that new freight car designs have to meet. These include performance limits for lateral stability on tangent track (hunting), operation in constant curves, spiral negotiation, cross level variation (twist and roll), surface variation (pitch and bounce), alignment variation on tangent track (yaw and sway) and alignment, gauge, and cross level variation in curves (dynamic curving). These tests use the ratio of lateral to vertical (L/V) forces exerted by the wheelset onto the rail, accelerations, degrees of roll and loading percentages to evaluate bogie performance. Among these criteria, the L/V criterion constitutes the most widely used performance metric in bogie testing. This makes intuitive sense since the wheelset is the component which connects the bogie to the track structure. The forces can be used in different combinations, as an individual wheel (L/V), axle sum Eq. (1) or truck side Eq. (2) ratio

$$Axle\ Sum\ \frac{L}{V} = \left[\frac{L}{V}\ (left\ ) + \frac{L}{V}\ (right\ )\right] \qquad (1)$$

$$Truck\ Side\ \frac{L}{V} = \left[\frac{\sum L\ (truck\ side)}{\sum V\ (truck\ side)}\right] \qquad (2)$$

Standard features of the modern rail car wheel, such as a flange and taper, have not always been part of the wheel. Figure 2 shows the two mentioned features on a wheelset.

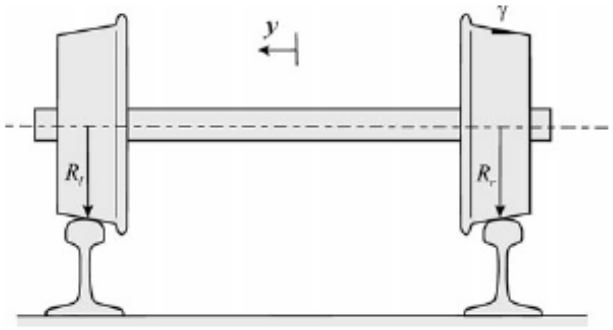


Figure 2. Wheelset in equilibrium position

Their invention, especially taper, can be credited to the need for improved guidance and proper curve negotiation. When the wheelset negotiates a curve, the outer rail follows a larger radius of curvature than the inner rail. This requires the outer wheel to travel a longer distance than the inner wheel. As the wheelset rotates with a constant angular velocity, one of the wheels or both wheels will slip. The slip can be reduced if the rolling radii of the two wheels are allowed to vary during the wheel motion. This change in the rolling radius is accomplished by using the tapered wheel profile. As the wheelset negotiates a curve, the wheelset will move laterally in the direction of the outer rail. Consequently, the outer wheel will have a larger rolling radius and higher velocity in the longitudinal direction as compared to the inner wheel. This reduces the slip and wear, and leads to better curving behavior (Shabana, Zaazaa, & Sugiyama, 2010). However, an inevitable side effect of the taper is the wheelset's inherent tendency to oscillate laterally. In 1883 Klingel (Klingel, 1883) derived the formula for this kinematic oscillation by relating wheel taper γ, wheel radius R_0, and distance between the wheel contact points *G*. Under perfect conditions on tangent track, the wheelset is centered with $y = 0$ and $R_l = R_r = R_0$. When the wheelset is laterally perturbed in the y-direction, the wheel taper will cause a decrease in radius for one wheel while the other wheel's radius increases. The combined difference ΔR in radii

$$\Delta R = y\gamma \qquad (3)$$

results in a difference in wheel velocities on the same axle and is reacted by a yawing motion of the wheelset as shown in figure 3.

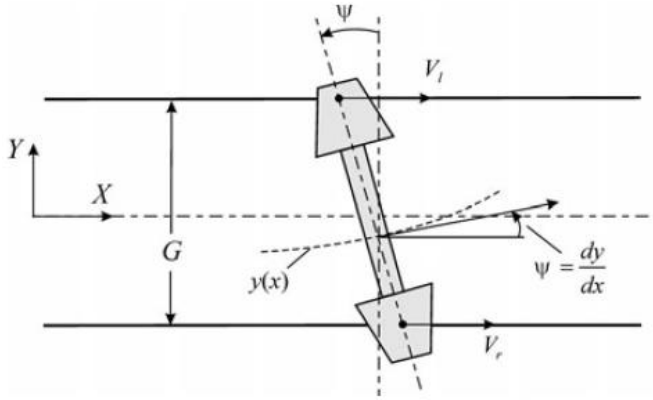


Figure 3. Hunting oscillation

In severe cases the wheelset will make flange contact with the rail in each oscillation as it "hunts" for its equilibrium position. For the same reason, this motion is commonly referred to as "hunting". The yaw motion is characterized by the yaw angle Ψ of the wheelset. In (Klingel, 1883) the underlying oscillatory motion of the wheelset was shown to be

$$\ddot{y} = \frac{-2R_0\omega^2 y\gamma}{G} \quad (4)$$

The solution of Eq. (4) is of the form

$$y = A\,sin(\omega_n t + C) \quad (5)$$

where A and C can be determined through initial conditions and ω_n is the natural frequency of the mechanical system.

$$\omega_n = V\sqrt{\frac{2\gamma}{R_0 G}} \quad (6)$$

Equations (5) and (6) are generally known as Klingel's Formulas (Klingel, 1883; Wickens, 1998) and describe the lateral oscillation of the wheelset due to the taper. The situation in which the taper of the wheels allows a bogie to negotiate a curve is the ideal for a perfectly aligned system. However, gradual wear from revenue service reduces this ability over time and affects bogie performance as a whole (Sawley, Urban, & Walker, 2005; Sawley & Wu, 2005). In addition to wheel wear, many other factors influence bogie performance. These include reduced warp restraint caused by worn suspension components, reduced rotational resistance caused by worn side bearings and manufacturing/reconditioning flaws such as mismatched side frames. Figure 4 shows four common misalignment faults of the bogie. In the case of rotational resistance it is worthwhile to note that a reduction decreases lateral stability but an increase worsens curving performance.

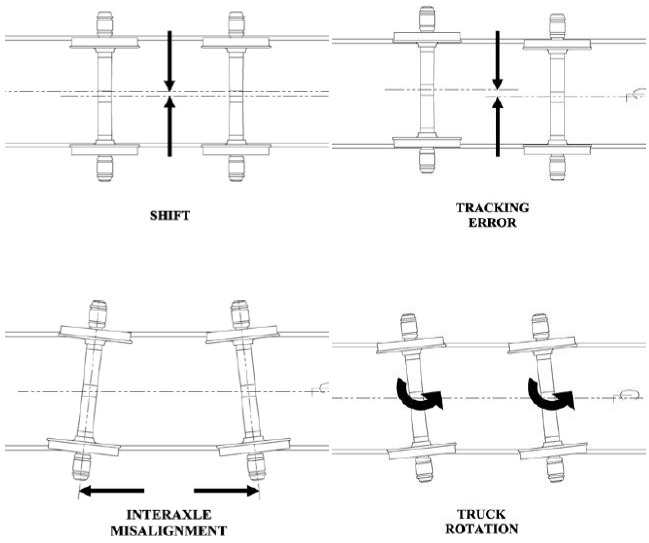


Figure 4. Bogie System Failure Modes

It is easy to see how each of the above mentioned fault conditions affects the wheelset alignment and triggers changes in the lateral and vertical forces of the wheels on the rail.

Failure modes of the rail car bogie system are generally defined as a decrease in performance and not a complete breakdown, as may be the case for other machinery. The industry relies heavily on wayside equipment for the detection of these deteriorated bogie components (Zakharov & Zharov, 2005). Different types of wayside equipment exist for detecting deteriorated parts on freight rail bogies. The two most relevant types for rail car bogie performance are Truck Performance Detectors (TPD) and Truck Hunting Detectors (THD). Both of these detectors consist of instrumentation which is added to the track to measure the lateral and vertical forces that rail car wheels exert on the track. TPDs achieve this through instrumentation of two reverse curves with strain gauges to measure the wheel lateral and vertical forces and wheelset angle of attack during curving. THDs are placed on tangent track and instrumented with strain gauges to measure wheelset hunting. Currently, approximately 15 TPDs and 172 THDs are in service across the North American rail network. The difference in their numbers stems from two reasons. First, TPDs are more expensive and more difficult to set up due to their two reverse curve requirement. Second, THDs are usually setup in conjunction with Wheel Impact Load Detectors (WILDs) as an additional functionality, adding less to the overall cost than a standalone TPD system. However, it is commonly accepted in the industry that TPD alerts are more worthy of repairs than THD alerts as they generally relate to a broader spectrum of root causes.

2. BOGIE PERFORMANCE CRITERIA

As mentioned previously, the Association of American Railroads Transportation Technology Center, Inc. (AAR/ TTCi) has established a set of design validation criteria for the quantification of bogie system performance through track testing. Although the tests consist of both static and dynamic requirements, this study will focus on dynamic requirements only. The dynamic requirements are divided into tests for smooth, unperturbed track and geometrically varying, perturbed track. The perturbed track tests are designed to excite vehicle dynamic modes historically associated with poor performance. The majority of the tests are evaluated by comparing wheel L/V force results against threshold limits per AAR MSRP C-II Chapter 11. Table 1 lists the criteria for these test regimes. As mentioned before, the most frequently used criterion of bogie performance (wheel L/V forces) comprises 9 out of the 21 requirements. This is followed by the percent load requirements (6) and acceleration based requirements (4). This shows that the industry has a historical affinity towards evaluating bogie performance by means of wheel L/V forces.

Table 1. AAR MSRP C-II Chapter XI Dynamic Performance Requirements

Test Regime	Criterion	Limit	
Hunting (empty)	Max. lat. Acc	1.5	[G]
	σ lat. Acc.	0.13	[G]
Constant Curving	95th perc max wheel	0.8	L/V
	95th perc max axle sum	1.5	L/V
Spiral Negotiation	Min. vert. load	10	[%]
	Max wheel	1.0	L/V
	Max axle sum	1.5	L/V
Twist/Roll	Max. roll	6	[°]
	Max axle sum	1.5	L/V
	Min. vert. load	10	[%]
	Dyn. augment acc.	1.0	[G]
	Loaded spring cap max.	95	[%]
Pitch/Bounce	Min. vert. load	10	[%]
	Dyn. augment acc.	1.0	[G]
	Loaded spring cap. max.	95	[%]
Yaw/Sway	Max. truck side	0.6	L/V
	Max axle sum	1.5	L/V
Dynamic Curving	Max wheel	1.0	L/V
	Max axle sum	1.5	L/V
	Max roll	6	[°]
	Min. vert. load	10	[%]

The unperturbed track tests include:

- *Lateral Stability on Tangent Track (Hunting):* hunting is the transfer of energy from forward motion into sustained lateral oscillations of the axle between the wheel flanges.
- *Operation in Constant Curves:* This tests the satisfactory negotiation of track curves. The resulting forces between wheel and rail have to be safe from any tendency to derail.
- *Spiral Negotiaion:* This tests satisfactory negotiation of spirals leading into and out of curves. The tests are required to show an adequate safety margin from any tendency to derail, especially under reduced wheel loading.

The perturbed track tests include:

- *Varying Cross-Level:* This tests the satisfactory negotiation of oscillatory cross-level excitations which may lead to large car roll and twist amplitudes. The tests have to show an adequate margin from any tendency to derail.
- *Surface Variation*: This tests the satisfactory negotiation of the car over track that provides an oscillatory excitation in pitch and bounce. A safety margin from any tendency to derail has to be shown.
- *Alignment Variation*: This tests the satisfactory negotiation of the car over track with misalignments that provide excitation in yaw and sway. A safety margin from any tendency to derail has to be shown.
- *Alignment, Gauge, Cross-Level Variation in Curves:* This tests the satisfactory negotiation of a combination of misalignments at low speeds. A safety margin from any tendency to derail has to be shown.

3. MODEL-BASED SIMULATIONS VS DATA DRIVEN DIAGNOSTICS

In recent years, the topic of advanced modeling techniques to supplement experiments such as the tests outlined above has received increased attention. In (Li & Goodall, 2004) a model-based approach is presented which derives theoretical knowledge from a mathematical model. Contrary to this method, data-driven approaches are used where mathematical models are unavailable and heuristic strategies have made solutions available. The authors argue in favor of a model-based approached, but steer their study away from complex non-linear simulation models. In the case of (Li & Goodall, 2004) this is permissible since it is assumed that the bogies in the study are passenger rail bogies with less non-linear effects, such as dry friction damping, stick-slip effects and clearances, than freight rail bogies (Iwnicki, 2006). The authors also mention the difficulties in generating fault accentuated signals (residuals) for fault detection and isolation purposes. Generally, a trade-off between accuracy and (computational) expense has to be considered when a realistic model is the goal. The alternative is to simulate hard faults, as the authors did in (Li & Goodall, 2004), even though this approach neglects gradual deterioration. Typical data-driven approaches usually focus more on gradual deterioration effects to establish cause and effect relationships. In both (Li & Goodall, 2004) and (Tsunashima & Mori, 2010) the proposed methods are tested only in simulation which is yet another drawback. Contrary to the opinion in (Li & Goodall, 2004) the best approach to be considered should be a combination of analytic simulation and experimental work. This is demonstrated in (Pogorelov, Simonov, Kovalev, Yazykov, & Lysikov, 2009) where the authors achieve this by using a multibody dynamics simulation package first to model the suspension and then validate their findings in a series of full scale experimental tests.

On the opposite end of the spectrum, purely empirical studies have been completed to determine root causes of suspension faults. In this type of study data is systematically collected to reflect failures as they appear in the field under revenue service conditions. In (H. M. Tournay & Lang, 2007; H. M. Tournay, Lang, & Wolgram, 2006) data from TPDs was analyzed and bogie systems which generated alerts were identified. Since the correlation between age and performance is well understood, old bogies with lowered warp restraint or mismatched side frames (due to reconditioning) were expected and not subject of the studies. The bogie systems with no obvious faults, which were expected to perform well, yet triggered an alert, were the

main subjects of both studies. The studies took a multitude of factors into consideration, including car maintenance history, TPD metrics (truck gauge spreading force, truck warp factor etc), truck parts/condition into account and identified potential root causes for poor performance. (H. M. Tournay, Lang, & Wolgram, 2006) concluded that side bearing malfunction and car body twist had caused line contact in the center bowl, and (H. M. Tournay & Lang, 2007) concluded that high bogie to carbody rotational resistance due to out of tolerance side bearings and high friction in the center bowl had triggered the truck performance detector alarms. Evidently, a purely data-driven analysis of wayside detector data intended to provide actionable results is very different from a model based technique to predict suspension failure based on simulated acceleration data. Empirical data is reflective of faults encountered in the field but may be difficult to interpret initially until repeated patterns can be systematically observed and attributed to their root causes. In contrast to this, model based approaches provide simulated data in which a single variable can be changed while others are held steady to isolate the root cause of a failure. The complexity and accuracy of a simulation strongly influences the applicability of results found in this manner.

In between a theoretical model based and data-driven approach fall data-driven techniques with advanced sensors but without mathematical models (Sunder, Kolbasseff, Kieninger, Rohm, & Walter, 2001). These methods present an interesting alternative as they are more practical than the model based approaches, and hence more applicable. However, the lack of a mathematical model underutilizes available simulation methods to improve accuracy either for sensor placement or algorithm and sensor threshold design.

The differences in the three presented approaches highlight the issues any condition based monitoring or predictive maintenance based approach faces.

3.1. Data-Driven Interpretation of Model-Based Simulation Data

The above presented model-based approaches do not outline how their goal of condition based maintenance should be achieved in practice. Implementation issues such as power on freight rail cars, reliability in harsh environments, feasibility and wireless communication remain entirely untouched. If these deficiencies were added to a model based approach, it could be a more viable solution in terms of an industrial application. An understanding of the faults, the maintenance practices, and operating environment can significantly strengthen conclusions obtained from the analysis of a theoretical bogie model and lead to results more reflective of industry practices. This paper is proposing the fusion of these two approaches to implement a system for data-driven based interpretation of model based data of railway bogie performance.

The key for this proposal is to devise a representative model of a freight rail bogie that is adequately detailed and not too complex to be computationally solvable. (Fujie & True, 2003) and (Pogorelov et al., 2009) used simulations with 19 rigid bodies and triple digit degrees of freedom models. These are significant numbers as they show the complexity of modeling the conventional North American three piece bogie. An investigation of which aspect of the bogie model would be most beneficial to model in higher detail to achieve the goal of fault simulation is recommended. Typically, the suspension system of the bogie is of the highest relevance amongst all bogie components. The suspension system of a freight rail bogie is made up of two subsystems. These are the primary suspension which consists of the adapter and adapter pad at the pedestal seat in the side frame and the secondary suspension which consists of the spring nest and friction wedges inside the side frame. One possible focus for the modeling efforts could be the secondary suspension of the bogie, as this is the main component which reacts the dynamic forces from the wheels on the rest of the bogie. Warp of the bogie system, resulting from worn secondary suspension components such as friction wedges could be considered a target fault. As mentioned in the introduction, bogie warp is a condition under which the friction wedges fail to resist the longitudinal shift of the side frames which results in misalignment. The misalignment rotates the wheelsets such that they exert a larger than normal track gauge spreading force onto the track in curves. Figure 5 shows the alignment of the wheelsets under conditions of a warped bogie.

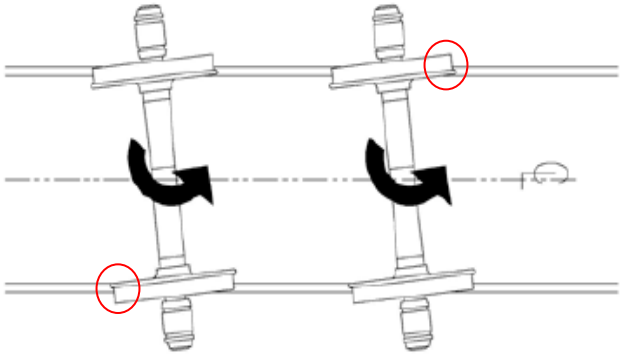


Figure 5. Wheelset alignment under warped bogie conditions

The red circles in figure 5 show where the increased forces would react with and potentially damage the track. Under lateral instability conditions (for loaded cars) on tangent track this fault would contribute to the development of hunting oscillations. It can be expected that symptoms of this fault will be discernible in the longitudinal acceleration signal from the side frames. An adequate method to iterate measurement responses towards deterioration should be implemented in the model. Measuring the response of bogie

components in terms of displacements and accelerations, would allow the creation of meaningful thresholds and the selection of the most beneficial location on the bogie for sensor placement.

Another interesting fault for the proposed method is hunting. Hunting was explained in the introduction as the lateral oscillatory motion of the bogie system, which is initiated by the wheel taper. It worsens over time as the wheel profile wears hollow and as a result the lateral oscillations increase in magnitude when the rail vehicle enters instability on tangent track. It can be expected that symptoms of this fault will be discernible in the lateral acceleration signal from the side frames, bearing adapters and rail car body. MSRP C-II Chapter 11 specifically mandates the use of worn wheel profiles for the hunting tests described above. The mandated (KR) profile is formalized as an approximation for a wheel profile after 100,000 miles of revenue service. Figure 6 shows the change in the profile from a new to a KR worn wheel.

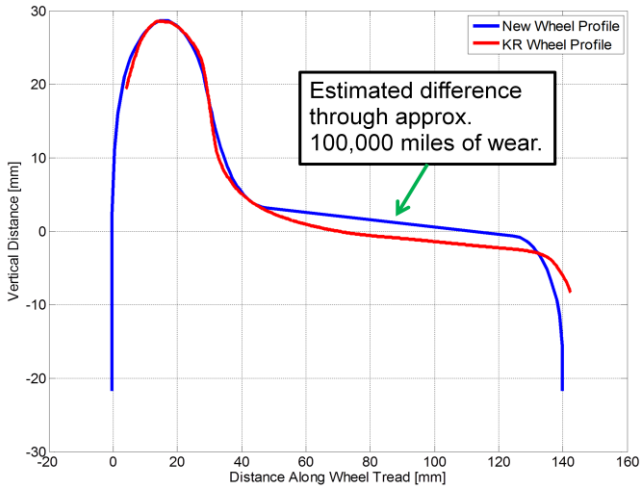


Figure 6. New wheel profile vs worn KR wheel profile

This fault mode is particularly interesting because MSRP C-II chapter 11 specifies acceleration levels as thresholds and not L/V ratios as it does for most of the other bogie performance tests. This makes the translation of regulatory requirements into actionable thresholds directly possible. Simulation results from the model will add the relationship of the oscillation severity to the wear of the wheel profile and potentially other root causes. These two examples show how the proposed method can be expanded and applied to additional bogie faults.

4. FIELD TEST

A first set of tests was conducted at Transportation Technologies Center, Inc. (TTCI) in Pueblo, CO. TTCI, a subsidiary of the Association of American Railroads, is a transportation research and testing organization. TTCI offers a wide range of tests for rail applications on their seven test tracks.

4.1. Field Test Setup

One of these tracks, the Railroad Test Track (RTT), is a 13.5-mile loop with four 50-minute curves and a single 1-degree, 15-minute reverse curve. Maximum speed is 165 mph and all curves have 6-inches of superelevation (difference in rail height on the same section of track - especially relevant in curves to maintain stability). The primary purpose of this track is high speed stability testing which is well suited for exciting lateral vehicle dynamic modes. The selection of lateral instability testing was based on two reasons: the first being that it is one of only two tests in MSRP C-II Chapter 11 which evaluate performance criteria as a quantity of acceleration in *G* and secondly, the industry's interest in modifying this specific requirement from currently empty cars to loaded cars. The increased interest in this particular instability mode is related to the introduction of higher load bogies as shown earlier in this paper. The higher car loads have resulted in wagon bodies with higher yaw/roll moments of inertia that react with relatively low warp restraint leading to coupled oscillatory resonance at speeds as low as 47 mph (H. Tournay, Wu, & Wilson, 2009). The extension of lateral instability tests is likely to affect product development and Mean-Time-To-Failure (MTTF) requirements, and as such poses a particularly well-suited example for an application of condition monitoring strategies.

For this study, one of the 50-minute (0.8 degree) curves with 6-inches superelevation was used to accelerate the train to target speeds, ranging from 40 mph to 80 mph. Figure 7 shows the profile of the segment of the RTT track that was used.

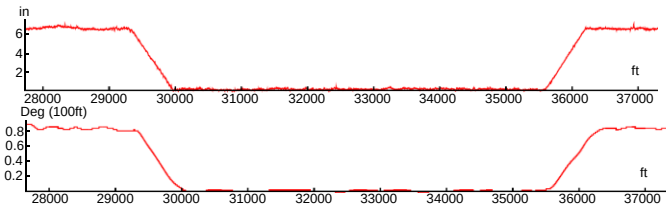


Figure 7. Test segment of RTT track

The upper graph shows the superelevation and the bottom graph shows the curvature. Once the target speed was reached, data acquisition systems began to measure the lateral and vertical accelerations at two sensor locations on the rail car body. Figure 8 shows the sensor locations at the A- and B-end on the loaded hopper car. The triangles indicate where the accelerometers were installed on the test car. Red indicates the accelerometers that were mounted near the roof of the car and green shows accelerometers on the deck above the bogie center location. The

instrumentation of the test car was in accordance with MSRP C-II Chapter 11 rules for trackworthiness testing of new freight car designs. As previously mentioned, per AAR rules, hunting is quantified as the peak to peak magnitude and standard deviation of the lateral acceleration on the deck above the center of the bogie. The two additional accelerometers (red in figure 8) were added to the test setup to measure lateral acceleration at the top of the rail car body.

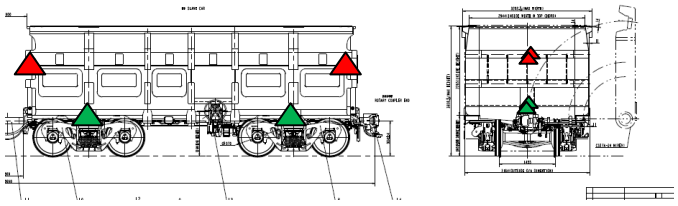


Figure 8. Instrumentation overview for loaded hopper car

Since the rail car body can be assumed to be rigid the extended moment arm between the center of rotation and measurement location at the top provides more pronounced acceleration which can be analyzed in correlation to the lower deck location. Additional signal processing requirements per the AAR rules were followed.

4.2. Field Test results

The field tests led to a number of significant results. Figure 9 shows the power spectral densities of each run's time series data from the rail car's top A-end location. It can be observed that a distinct resonant frequency becomes detectable above 55 mph and that the resonance is located between 2.0 and 3.0 Hz, depending on the speed of the test run. This is not a coincidence as it is well known in the industry that hunting occurs in this frequency range.

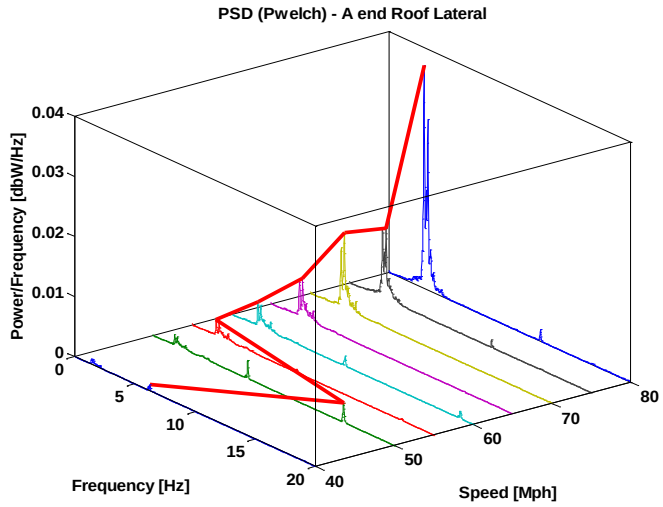


Figure 9. Frequency domain data between 40 and 80 mph

Furthermore, this frequency range also correlates to that of the kinematic analysis in the introduction and can be regarded as the propagated vibration of the wheelset's side to side oscillation in which the wheel flange contacts the rail. The finding of this result is significant because it shows that when factors such as wheel taper and lading are controlled so that they favor excitation of a dynamic failure mode, accelerations indicative of this failure can be measured. Moreover, the progressively increased test speeds show the gradual increase of the oscillatory power in frequency domain. The increased oscillatory power at the roof of the car body versus the sill location can be observed in figure 10. There, the 80 mph test run data is shown in four different locations and it can be observed that the roof and sill follow similar trends with different magnitudes.

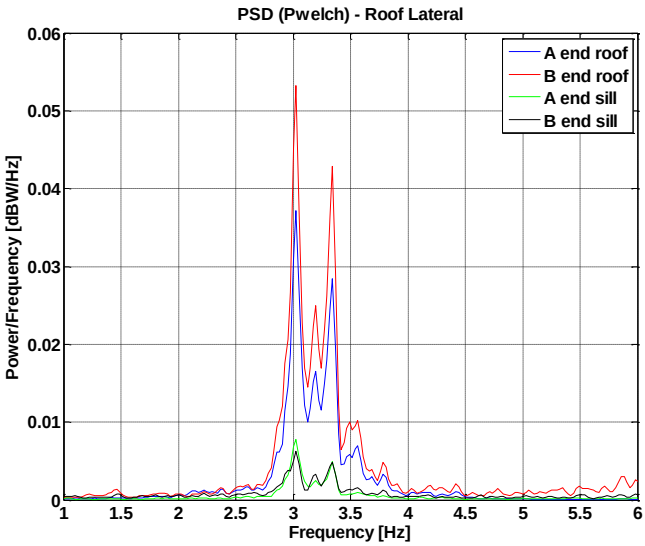


Figure 10. Comparison of roof vs sill location at 80 mph

5. DISCUSSION

It was shown in the field test section that actionable information could be obtained from accelerometers in the sill or roof locations of the rail car. This first test can be assumed as a proof of concept for expansion of the outlined monitoring strategy to the following additional bogie faults, historically associated with certain component failures:

- Bogie Misalignment: figure 4 in the introduction showed four different misalignment faults for bogies. Having various root causes (H. M. Tournay, Lang, Wolgram, & Chapman, 2006) these misalignments lead to forces resulting from the complex, dynamic interactions of the bogie parts and track. Identification of interactions such as warp restraint and angle of attack and the effect an increase or reduction would have on the dynamic behavior of the bogie system is proposed.
- Spring Nest: faulty operation of this suspension component is coupled to the vertical motion of the bolster and anomalies could be detectable if there is a

significant change in the displacement when this component wears.

- Side Bearings: are intended to support the even distribution of the lading and prevent hunting. If contact forces are too high, the rotation of the car body against the bogie can be inhibited leading to high curving forces. If they are too low, lateral oscillations will not be adequately resisted.
- Wheels: this fault can be quantified by wheelset lateral oscillations as they occur when wheels are worn hollow and begin to lose their self-centering abilities as outlined in the kinematic analysis.

For the first three of the above described faults a triaxial accelerometer would be a suitable sensor package to identify the faults. The longitudinal axis would sense side frame displacements due to bolster rotation, the vertical axis would sense bolster vertical displacements and the lateral axis would sense lateral oscillations such as bogie hunting. For the last fault, wheelset displacements, the best acceleration axis would be the lateral axis.

To detect these faults the selected sensor package would be placed on the bogie. Multiple locations meet the requirements outlined above and could work but should be investigated in simulations and field testing to confirm applicability. Three particular locations are of high interest: 1. Either end of the side frame, 2. Either end of the bolster and 3. Bearing adapter locations. Additional knowledge can be gained by placing accelerometers on the car body, especially if yaw/roll coupled instability modes of the car body are of interest. Simulating the dynamic modes with a model and supplementing the findings with a field test would provide a better understanding of which location is preferable and provides higher accuracy in detecting these faults.

To create actionable thresholds it would be furthermore of interest to relate currently existing TPD alarm levels to acceleration limits. TPDs classify bogies as bad actors based on force and angle of attack based TPD data. The criteria for this are either two events exceeding the forces shown in figure 11 within 12 months or two Lead Axle High Rail L/V values of 1.05 also within 12 months. Both of these requirements were established in parallel to MSRP C-II Chapter 11 and are outlined in detail in (H. M. Tournay, Lang, Wolgram, et al., 2006). Multibody simulation packages are able to estimate these wheel lateral and vertical forces as part of a simulation. One issue the authors mention is the intermittent behavior of TPDs during successive passes of the same car. It has proven to be a major obstacle to the interpretation of TPD data. This is yet another aspect in favor of the proposed monitoring approach.

For THDs the condemning criteria are either two events with a Salient Hunting Index above or equal to 0.35 or a single Salient Hunting Index above 0.5. Hunting is investigated in (H. M. Tournay, Wu, & Wilson, 2008) with respect to its occurrence under loaded car conditions. This is relevant as it directly pertains to the pending rule change to extend empty car criteria to loaded car criteria. Investigation of factors such as adapter pad (primary suspension) and wheel profile combinations resulted in concluding that loaded car hunting is a resonant coupling between the yaw oscillation of the wheelset and natural frequency of rail car body in a yaw mode that includes in-phase body roll motion.

Table 2. TPD Truck gauge spread force (TGSF) limits

TGSF (kips)	Site Curvature (degrees)
28	≤ 4.0
33	$\geq 4.0 < 5.0$
38	$\geq 5.0 < 6.0$
43	$\geq 6.0 < 7.0$
48	$\geq 7.0 < 8.0$
53	$\geq 8.0 < 9.0$
58	≥ 9.0

From a component perspective it primarily depends on frictional warp properties, adapter pad stiffness and taper wear of the wheelsets. A meaningful combination of these fault modes and hierarchical structure for which to monitor first shall be derived from these initial findings.

6. CONCLUSION

Problems in monitoring the condition of the standard North American three piece bogie were outlined in this study and a strategy to attack these from a combined data-driven and analytic simulation approach was presented. An overview of bogie performance standards from a regulatory perspective and existing technologies that are currently in use in railroad revenue service was provided. Challenges that these technologies pose in terms of implementation effort, preventive action effectiveness, and faulty component identification were presented.

A field study presented initial results of an investigation of lateral instability and how these results can be used to detect gradual wear in components that are tied to a particular fault mode. The addition of a model to simulate these failures prior to field testing was proposed and would enable researchers to make decisions about locations for sensor placement and thresholds. Finally, currently used

performance parameters for the two dominant monitoring technologies were presented and it was outlined how these performance parameters could be 1) linked to components associated with the performance parameters, 2) adopted in a condition monitoring strategy to reflect the existing performance standards. As an extension of this strategy the failure mode of loaded car hunting was presented as an example in which application of the proposed strategy is particularly sensible, as the determining performance factor can be directly linked to the regulatory standard and sensor measurements.

REFERENCES

AAR. (2007). Design, Fabrication, and Construction of Freight Cars *Manual of Standards and Recommended Practices C-II* (Vol. [M-1001]).

Fujie, X., & True, H. (2003, 22-24 April 2003). *On the dynamics of the three-piece-freight truck.* Paper presented at the Rail Conference, 2003. Proceedings of the 2003 IEEE/ASME Joint.

Iwnicki, S. (2006). *Handbook of railway vehicle dynamics*: CRC Press.

Klingel, W. (1883). Über den Lauf der Eisenbahnwagen auf gerader Bahn. *Organ für die Fortschritte des Eisenbahnwesens, 20*, 113-123.

Li, P., & Goodall, R. (2004). *Model-based condition monitoring for railway vehicle systems.* Paper presented at the Proceedings of the UKACC international conference on control, Bath, UK.

Pogorelov, D., Simonov, V., Kovalev, R., Yazykov, V., & Lysikov, N. (2009). *Simulation of Freight Car Dynamics: Mathematical Models, Safety, Wear*. Paper presented at the International Conference on Recent Advances in Railway Engineering (ICRARE), Tehran, Iran.

Sawley, K., Urban, C., & Walker, R. (2005). The effect of hollow-worn wheels on vehicle stability in straight track. *Wear, 258*(7–8), 1100-1108. doi: http://dx.doi.org/10.1016/j.wear.2004.03.058

Sawley, K., & Wu, H. (2005). The formation of hollow-worn wheels and their effect on wheel/rail interaction. *Wear, 258*(7–8), 1179-1186. doi: http://dx.doi.org/10.1016/j.wear.2004.03.029

Shabana, A. A., Zaazaa, K. E., & Sugiyama, H. (2010). *Railroad vehicle dynamics: A computational approach*: CRC Press.

Sunder, R., Kolbasseff, A., Kieninger, A., Rohm, A., & Walter, J. (2001). *Operational experiences with onboard diagnosis system for high speed trains.* Paper presented at the Proceedings of the World Congress on Rail Research.

Tournay, H., Wu, H., & Wilson, N. (2009). A Review of the Root Causes for Loaded Car Hunting, Technology Digest TD-09-014: AAR, TTCI, Pueblo, CO.

Tournay, H. M., & Lang, R. (2007). History and Teardown Results of Five Loaded Coal Cars Identified as Poor Performers while Passing accorss a Truck Performance Detector *R-985*. Washington, DC: Association of American Railroads/ Transportation Technologies Center, Inc.

Tournay, H. M., Lang, R., & Wolgram, T. (2006). History and Teardown Results of a Coal Car Identified as a Poor Performer while Passing Loaded Across a Truck Performance Detector *R-976*. Association of American Railroads50 F Street, NW Washington, DC 2001.

Tournay, H. M., Lang, R., Wolgram, T., & Chapman, S. (2006). Interpreting Truck Performance Detector Data To Establish Car And Truck Condition *R-977* (Vol. R-977). Association of American Railroads50 F Street, NW Washington, DC 2001.

Tournay, H. M., Wu, H., & Wilson, N. (2008). Investigation into the Root Causes for Loaded Car Hunting *R-995*. Association of American Railroads 50 F Street, NW Washington, DC 2001.

Tsunashima, H., & Mori, H. (2010, 27-30 Oct. 2010). *Condition monitoring of railway vehicle suspension using adaptive multiple model approach.* Paper presented at the Control Automation and Systems (ICCAS), 2010 International Conference on.

Wickens, A. H. (1998). The dynamics of railway vehicles—from Stephenson to Carter. doi: 10.1243/0954409981530805

Zakharov, S. M., & Zharov, I. A. (2005). Criteria of bogie performance and wheel/rail wear prediction based on wayside measurements. *Wear, 258*(7–8), 1135-1141. doi: http://dx.doi.org/10.1016/j.wear.2004.03.025

BIOGRAPHIES

Parham Shahidi is a Project Engineer in the Bogie Systems Engineering group at Amsted Rail Company, Inc. He holds a BS degree in Mechanical & Process Engineering from TU Darmstadt in Germany and a PhD in Mechanical Engineering from Virginia Tech. Parham has been working in the area of railroad engineering since 2007 starting with the development of a fatigue estimation system for conductors in graduate school. In 2011 he joined Amsted Rail where he has been working on rail vehicle dynamics and bearing condition monitoring projects. He has published more than 10 technical articles related to his research and is a board member of the German Engineers Association (VDI) in North America.

Dan Maraini is the Manager of Bogie Systems Engineering for Amsted Rail Company, Inc. He leads a team of Project Engineers focusing on condition monitoring solutions for

freight railcars. Dan has been a part of Amsted Rail since 2008. In various positions, he has worked on wireless sensing, remote asset monitoring, and condition monitoring for freight rail applications. Dan is currently a PhD student in the department of Mechanical Engineering at Villanova University. His research focus lies on machinery diagnostics. He holds a MS in Mechanical Engineering from Villanova University and a BS in Physics from West Chester University of PA.

Brad Hopkins is a Project Engineer in the Bogie Systems Condition Monitoring group at Amsted Rail. He holds a BS, MS, and PhD in Mechanical Engineering from Virginia Tech. He has been working on track and railcar condition monitoring since 2010, with a focus on accelerometer-based monitoring and algorithm development. His current work includes broken rail detection, wheel defect monitoring, and end-of-car system monitoring. His additional research interests are vehicle dynamics, modeling, and simulation, controls, and vibration analysis.

Andrew Seidel is a Project Engineer in the Bogie Systems Engineering group at Amsted Rail Company, Inc. He holds a BS degree in Mechanical Engineering from Pennsylvania State University in State College, Pennsylvania. Andrew has been working in the area of railroad engineering since 2012 when he began working with end of car impact detection systems and railcar condition monitoring projects. He is a member of the American Society of Mechanical Engineers (ASME) and sits on the Mechanical Engineering Technical Advisory Board for Penn State University (Harrisburg campus).

Particle Filtering-Based System Degradation Prediction Applied to Jet Engines

Peng Wang and Robert X. Gao

Department of Mechanical Engineering, University of Connecticut, Storrs, CT, 06269, USA
pew12002@engr.uconn.edu
rgao@ engr.uconn.edu

ABSTRACT

This paper investigates a real-time fault detection and degradation prediction scheme for dynamical systems such as jet engines, based on Regularized Particle Filtering (RPF). Particle Filtering is a prognosis method for the prediction of state degradation and remaining useful life (RUL) due to its demonstrated performance in handling non-linear and non-Gaussian situations. RPF overcomes the problem of sample impoverishment among particles over the resampling process. Based on measured data from hybrid sensing and nonlinear models, which link system parameters and degradation state to the measurement, RPF has been applied to establishing a framework for both state and parameter estimation, to achieve prognosis at the component level. In addition, a modified system evolution model is proposed to track both exponential and transient types of system performance degradation. The developed method is evaluated using simulated data created with C-MAPSS, which contains measured parameters associated with engine degradation under nominal and varied fault types (fan, compressor and turbine) during a series of flights. The developed system-parameter estimation method is found effective in state estimation and degradation prediction in jet engines.

1. INTRODUCTION

In most cases real world data contain failure signatures but little to no information about the failure evolution or state degradation, thus driving the need for health monitoring, diagnosis of faults, system performance degradations and trend prediction for dynamic systems, such as jet engines. Several prevalent sensing and diagnosis techniques have been proposed in past decades for health management in jet engines, such as gas path analysis (Volponi, 2003), exhaust composition and gas path debris (Simon, Garg, Hunter, Guo & Semega, 2004). Gas path analysis (GPA) is one of the most popular techniques to quantify the thermodynamic performance of engines based on the hybrid sensing of temperature, pressure and other measurements. The approaches to establish the relationship between measurement and system state can be classified into two categories: data-driven and model-based. A data-driven approach requires a large amount of historical data for training and lacks generality (Peng, Dong & Zuo, 2010), while a model based approach takes advantage of merits of both physical knowledge and historical data information.

Depending on system types and noise assumptions, different methods including the Kalman filter (for linear system and Gaussian noise) (Kalman, 1960), the extended Kalman filter (for weak nonlinear system and Gaussian noise) (Julier & Uhlmann, 1997), and the particle filter (for nonlinear system and non-Gaussian noise) (Gordon, Salmond & Smith, 1993) can be applied to implement model based prognosis (Doucet & Johansen, 2009). Due to the stochastic and nonlinear nature of the engine system performance degradation, this paper presents a probabilistic degradation prediction method to achieve the diagnosis and prognosis at the component level by recursively updating the physical model with online measurement based on Regularized Particle Filtering (RPF), while RPF is proposed to overcome the sample impoverishment problem in the resampling stage of standard PF (Musso, Oudjane & Legland, 2001). Besides exponential degradation prediction, a modification of the state evolution model has been proposed to track transient changes in system state and parameters due to faults.

The rest of the paper is constructed as follows. Theoretical background of particle filtering and the modified system evolution model are introduced in Section 2, followed by the discussion of the system degradation model and thermodynamic measurement models of engines at the component level that are implemented in RPF based prognosis in Section 3. The effectiveness of the presented technique is demonstrated in Section 4, based on run-to-failure simulated data created with C-MAPSS. Finally, conclusions are drawn in Section 5.

Peng Wang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. FILTERING FRAMEWORK

In order to analyze and make inference about a dynamic system, the posterior probability density function (pdf) needs to be estimated and updated for the underlying system state, based on the availability of new measurements, in the Bayesian framework. The system model describing the evolution of the state (variables representing system performance degradation in this paper) with time and the measurement model relating observable noisy measurements to true state are not nonlinear in many dynamic systems. Particle Filtering, also referred as Sequential Monte Carlo (SMC) (Orchard, Cerda, Olivares & Silva, 2012), provides a numerical approximation for nonlinear system estimation, using a set of random samples (or particles) with associated weights to construct the pdf of a state (Gordon, 1993).

2.1. Regularized Particle Filtering

For the estimation of the underlying state in a nonlinear dynamic system, it is assumed the stochastic model of system evolution is known as:

$$x_k = f_k(x_{k-1}, w_{k-1}) \tag{1}$$

where $f_k : \mathbb{R} \quad \mathbb{R} \quad \mathbb{R}$ describes the state transition function from state x_{k-1} to x_k considering an order-one Markov process. w_{k-1} is the process noise representing uncertainty. The state is recursively estimated based on the measurements (Saha & Goebel, 2011):

$$z_k = h_k(x_k, v_k) \tag{2}$$

where $h_k : \mathbb{R} \quad \mathbb{R} \quad \mathbb{R}$ is the measurement function representing the relation between online measurements z_k and an unobservable degradation state x_k. v_k is the sequence of measurement noise.

In the Bayesian framework, estimation is fulfilled by recursively calculating the posterior pdf $p(x_k|z_{1:k})$ of the state given the noisy measurements $z_{1:k}$ (Wang, Wang & Gao, 2013). Taking into account the one-step Markov process, the pdf can be obtained using two stages: prediction and update, as shown in Eq. (3) and Eq. (4).

$$p(x_k \mid z_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{k-1}) dx_{k-1} \tag{3}$$

$$p(x_k \mid z_k) = \frac{p(x_k \mid z_{k-1}) p(z_k \mid x_k)}{p(z_k \mid z_{k-1})} \tag{4}$$

where $p(z_k|z_{k-1})$ is the normalizing factor which can be calculated as:

$$p(z_k \mid z_{k-1}) = \int p(x_k \mid z_{k-1}) p(z_k \mid x_k) dx_k \tag{5}$$

In particle filters, the posterior pdf is represented and approximated by a set of random samples or particles $\{ x_{1:k}^i$, i = 1, 2, …, N$\}$ and associated importance weights w_k^i. The weights are normalized with $\sum_i w_k^i = 1$. The integral operation in Eq. (3) is then approximated as the summarization of these random numbers as:

$$\begin{aligned} p(x_k \mid z_{k-1}) &= \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{k-1}) dx_{k-1} \\ &\approx \sum_{i=1}^{N} w_{k-1}^i \delta(x_{k-1} - x_{k-1}^i) p(x_k \mid x_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i p(x_k \mid x_{k-1}^i) \end{aligned} \tag{6}$$

where the total number of particles N can affect the accuracy of the represented probability distribution, and computational efficiency. In the update step, the weight of each particle is updated based on the likelihood of the observation z_k at time k as:

$$w_k^i \propto w_{k-1}^i p(z_k \mid x_k^i) \tag{7}$$

Similarly, the posterior probability distribution $p(x_{k+l}|z_k)$ in the l-step ahead prediction can be obtained as:

$$p(x_{k+l} \mid z_k) \approx \sum_{i=1}^{N} w_{k+l-1}^i p(x_{k+l} \mid x_{k+l-1}^i) \tag{8}$$

In constructing the particle filter, resampling is applied in every step to remove particles with small weights (justified by comparing the cumulative distribution function to a threshold within 0~1) and obtain equally weighted samples so as to avoid the degeneracy problem of the algorithm. After resampling, the weights of the new particle population are reset to $w_k^i = 1/N$. However, in the standard PF methods stated above, due to the fact that the samples are drawn from discrete distributions instead of continuous distributions, the problem of loss of diversity among the particles may arise. To overcome this problem, the Regularized Particle Filter (RPF) has been proposed. The fundamental idea is to change the discrete approximation to a continuous one of posterior pdf in the resampling stage with the rescaled kernel structure. The update process Eq. (4) becomes:

$$p(x_k \mid z_k) = \sum_{i=1}^{N} w_k^i K_h(x_k - x_k^i) \tag{9}$$

Where

$$K_h(x) = \frac{1}{h^{n_x}} K(\frac{x}{h}) \tag{10}$$

$K(\cdot)$ is the recalled kernel density and h is the kernel bandwidth, the selection of which is optimally related to the dimension of state n_x and the number of particles N.

2.2. System Model for Transient Degradation

System estimation includes state estimation and parameter estimation. In most cases, the parameters are included in the state transition function $f_k : \mathbb{R} \quad \mathbb{R} \quad \mathbb{R}$, and then it becomes the joint state and parameter estimation. For most dynamical models like the performance degradation, the parameters are assumed to be constant within in a small range and the artificial evolution law is adopted (Liu & West, 2001), then the state will decay in an exponential

way. However these state models do not consider the case of transient degradation due to faults, which would cause a transient change in both parameters and states (Daroogheh, Meskin & Khorasani, 2013). The idea to handle this problem proposed in this paper is to include the output prediction error or measurement innovation into the state evolution model.

If fault occurs between sampling time k and k+1, the parameters used to predict the state x_{k+1} and output z_{k+1} are assumed to be consistent with values in previous sampling times 1:k. Thus there will be transient change of the output prediction error between time k and k+1. The solution is to compare the cost function

$$J = E[\frac{1}{2}(z_{k+1} - \hat{z}_{k+1})(z_{k+1} - \hat{z}_{k+1})^T] \quad (11)$$

to a predefined threshold. Where $\hat{z}_{k+1}$ is the predicted output at time k+1. If the cost function exceeds the threshold, the state evolution model Eq. (1) becomes:

$$x_{k+1} = \mathrm{x}_k + \gamma_k u + w_k \quad (12)$$

where u is the unit step function and γ_k is the time varying gain related to the cost function J. The additional item $\gamma_k u$ is to track the state change due to failures.

3. MODEL FORMULATION

Gas path analysis relies on discernable changes in observable parameters to detect physical faults. The fundamental tenet underlying this approach is that physical faults occurring in components (fan, low/high pressure compressor and high/low pressure turbine) of engines induce a change in component performance (modeled as efficiency, flow capacity, etc.), which in turn produce observable changes in measureable parameters (temperature, pressure, speeds, etc.). This inverse relationship offers the approach for engine performance estimation (Volponi, 2003). In the implementation of fault detection and degradation trend prediction of engines at the component level, using the proposed estimation method, the efficiency of each component is considered as the state needing to be estimated from observable measurements.

The exponential behavior of the fault evolution or system performance degradation is common for all degradation models (Saxena, Goebel, Simon & Eklund, 2008). Thus, a generalized state evolution model in this paper is assumed as:

$$x_k = x_{k-1} - \exp(A_{k-1}\tau^{B_{k-1}}) + w_{k-1} \quad (13)$$

where A_{k-1} is the scaling factor and B_{k-1} is the time-varying factor determining the degradation rate at sampling k-1. τ is the sampling interval and w is the associated process noise. In the training stage, parameters A and B are estimated using RPF iteratively. In the prediction stage, the latest updated parameters assigned with each particle joint with state evolution model would provide the predicted states. Namely, the parameters stay constant in the prediction stage (Zhu, Yoon, He, Qu & Bechhoefer, 2009).

The nonlinear measurement equations that relate state (efficiency) and measurements for compressor and turbine (Moran & Howard, 2004) are listed as follows

$$CPR = P_{Cout} / P_{Cin}$$
$$T_{Cout} - T_{Cin} = \frac{T_{Cin}}{\eta_C}(CPR^{\frac{\gamma_C - 1}{\gamma_C}} - 1) \quad (14)$$

$$T_{Tout} - T_{Tin} = \frac{T_{Tin}}{\eta_T}((\frac{P_{Tout}}{P_{Tin}})^{\frac{\gamma_T - 1}{\gamma_T}} - 1) \quad (15)$$

where, T_{Cin}, T_{Cout}, T_{Tin} and T_{Tout} denote the temperature of the inlet and outlet of the compressor (low/high pressure) and turbine (low/high pressure), respectively, and P_{Cin}, T_{Cout}, T_{Tin} and T_{Tout} denote the temperature of the inlet and outlet of the compressor and turbine, respectively. CPR is the abbreviation of compressor pressure ratio. γ_C and γ_T denote the specific heat ratio of the compressor and turbine, which are assumed to be constant. η_C and η_T denote the efficiency of the compressor and turbine, which are also assigned as the state parameter to represent engine status.

Even if no fault occurs, the engine performance still decays in an exponential way, causing an accumulative efficiency loss of each component, which in turn is represented by discernable changes of observable measurements. Fig (1) gives an example of accumulative efficiency loss and corresponding measurement change of the high pressure compressor (HPC). More details on implementation of degradation trend prediction and transient decay detection using proposed diagnosis and prognosis method are discussed in the next section.

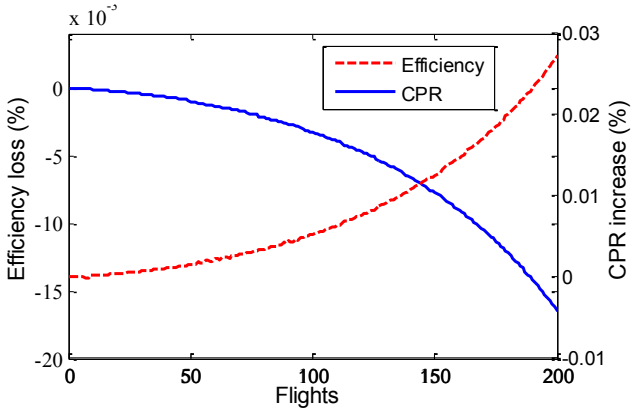


Figure 1. Accumulative efficiency loss and corresponding CPR increase of HPC

4. PERFORMANCE EVALUATION

To evaluate the performance of the proposed RPF based engine degradation prediction method, a set of high fidelity system level engine simulation data is used (Saxena, 2008). The data is created with a Matlab Simulink tool called C-MAPSS, designed to simulate normal and fault engine degradation over a series of flights. Each flight is a combination of a series of flight conditions with a reasonable transition period to allow the engine to change from one flight condition to the next. For the normal condition case, the engine is given an exponentially degrading fuel flow and efficiency profile, which denote the degradation of system performance. For fault condition cases, the engine is assigned one of five possible faults (fan, LPC, HPC, HPT and LPT) at a random flight. The fault is manifested by increasing the efficiency parameters degradation from the fault time point until the end of the simulation for the remaining flights. After a flight is simulated, a snapshot of all engine parameters is taken in the middle of cruise and applied to estimate engine state and predict the degradation trend.

In the learning stage, based on the state equations (denoted by Eq. (12) and Eq. (13)) and measurement equations (denoted by Eq. (14) and Eq. (15)), the state transition probability $p(x_k|x_{k-1})$ and measurement probability $p(z_k|x_k)$ can be obtained as *a priori*, then the posterior distribution function of efficiency state $p(x_{k+l}|z_k)$ can be predicted using the RPF. In the system equation, the model parameters A and B in Eq. (13) are modeled as probability distributions following the uniform distribution, to incorporate the stochastic property of the engine component degradation. The latest update of these two parameters helps construct the state transition probability $p(x_k|x_{k-1})$ and subsequently the degradation prediction. Fig. 2 shows an example of HPC efficiency degradation prediction based on the developed methods under a normal case (natural decay, no fault occurrence), using the information of the first 160 flights as the prior knowledge to predict the efficiency trend.

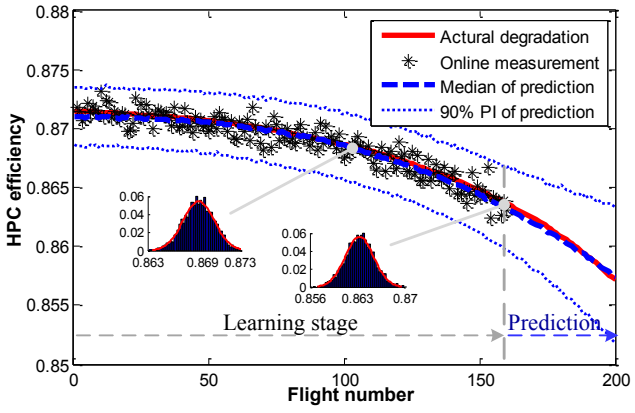


Figure 2. Predicted HPC efficiency degraded in an exponential way without external fault

Fig. 3 shows the HPC efficiency prediction under fault case, where the transient decay occurs at the 23rd flight by a 0.25% loss. Also, the information about the first 80 flights is taken as the prior knowledge for the proposed method to predict the efficiency evolution of the last 20 flights. The simulation result indicates that the proposed method can track the both exponential and transient types of system performance degradation. Because the cost function of estimated output at previous sampling time, as denoted by Eq. (11), is checked each step, there is a time delay for the estimation to track the transient change. Fig. 4 is the evolution of distribution of parameters A and B in Eq. (13). It is noted that the value of both parameters are consistent before and after the transient change. In addition, because there is no new information to update the parameters, they stay the same in the prediction stage.

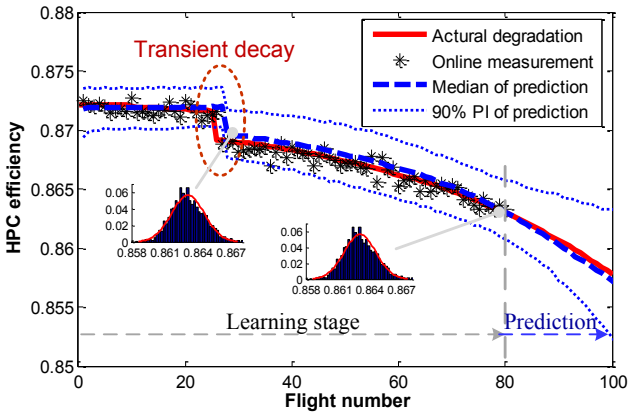


Figure 3. Predicted HPC efficiency with a transient decay under the effect of external fault

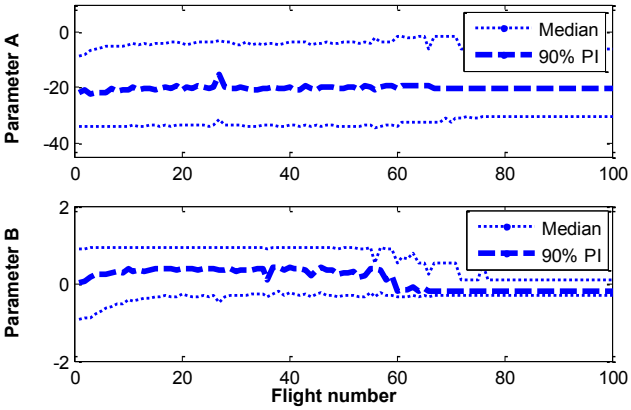


Figure 4. Evolution of distribution of parameters A and B for HPC efficiency estimation

Fig. 5 shows another example of LPT efficiency prediction in the fault case, where the fault occurs at 33rd flightFig. 6 is the corresponding evolution of parameters distribution.

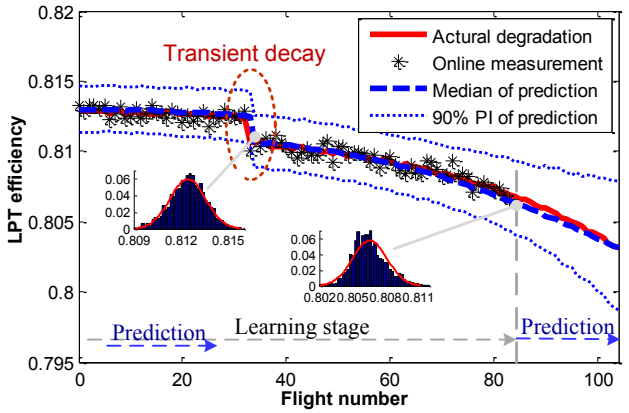


Figure 5. Predicted LPT efficiency with a transient decay under the effect of external fault

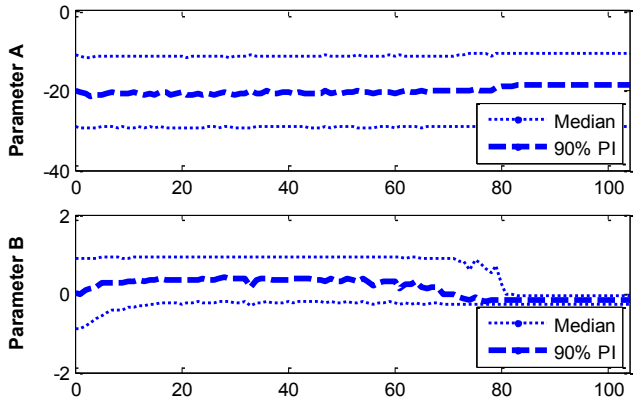


Figure 6. Evolution of distribution of parameters A and B for LPT efficiency estimation

To evaluate the effectiveness and robustness of proposed method on degradation prediction, Monte Carlo simulation is applied to derive the comprehensive simulation results. Each scenario has been run for 100 times. Mean and root mean square (RMS) of root mean square error (RMSE) of median prediction are listed in Table 1.

Table 1 Monte Carlo simulation result of proposed method

	Normal HPC	Fault HPC	Fault LPT
Mean	0.086%	0.1%	0.13%
RMS	0.097%	0.12%	0.14%

Extended Kalman filter (EKF) is selected here as the alternative method to compare with PF, while the results are shown in Fig. 7. Maximum likelihood (ML) integrated with EKF is adopted to estimate the unknown parameters in the state evolution model, based on which prediction is performed. It is found that prediction accuracy of PF is over EKF+ML, and the prediction accuracy of natural degradation over the mixed degradation.

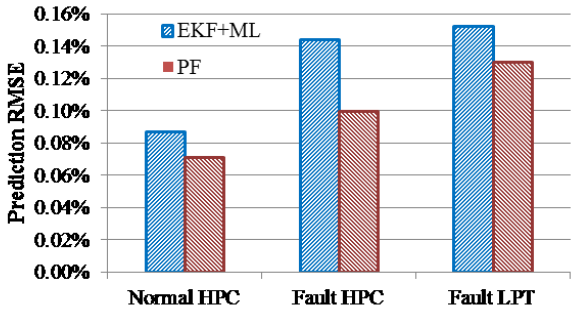


Figure 7. Performance comparison between PF and EKF

5. CONCLUSION

Particle Filtering has been investigated as a prognostic method for both state and parameter estimations in determining the efficiency degradation of jet engines as an example of dynamical system prognosis, at the component level. State estimator is modified by a cost function that compares the predicted measurements to updated measurements, and enables the tracking of transient decays in addition to exponential type of degradations. Simulated data sets including normal and fault cases generated by the C-MAPSS program have been used to evaluate the effectiveness of the developed algorithm for engine degradation state prediction, with quantified confidence intervals to manage uncertainty. In the three examples considered, the results indicate that the method can track transient changes within two steps, and the prediction error is less than 1%. Future research will investigate the robustness of the developed algorithm for different applications under different operational conditions, using experimental data.

ACKNOWLEDGEMENT

This research has been partially supported by National Science Foundation under grant CMMI-1300999 and CNS-1239030.

REFERENCES

Doucet A., & Johansen A. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. In Crisan D., & Rozovsky B., *The Oxford Handbook of Nonlinear Filtering*, (656-704). Oxford: Oxford University Press.

Daroogheh, N., Meskin, N., & Khorasani, K. (2013). Particle filtering for state and parameter estimation in gas turbine engine fault diagnostics. *Proceedings of 2013 American Control Conference*. June 17-19, Washington, DC, USA.

Gordon, N. J., Salmond, D., & Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F Radar and Signal Processing*, vol. 140(2), pp. 107-113. doi: 10.1049/ip-f-2.1993.0015

Julier, S., & Uhlmann, J.,(1997). A new extension of Kalman Filter to nonlinear systems. *Proceedings of 11th International Symposium on Aerospace/Defense sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*. July 28, Orlando, FL, USA. doi: 10.1117/12.280797

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35-45. doi: 10.1115/1.3662552

Liu, J., & West, M. (2001). Combined parameter and state estimation in simulation based filtering. In Doucet, A., Freitas, N., & Gordon, N., *Sequential Monte Carlo Methods in Practice*, (197-223). NY: Springer New York.

Moran, M., & Howard, N. S. (2004). *Fundamentals of Engineering Thermodynamics*. NJ: Wiely.

Musso, C., & Oudjane, N. (2001). Improving regularized particle filters. In Doucet, A., Freitas, N., & Gordon, N., *Sequential Monte Carlo Methods in Practice*, (197-223). NY: Springer New York.

Orchard, M.E., Cerda, M., Olivares, B., & Silva, J. (2012). Sequential Monte Carlo methods for Discharge Time Prognosis in Lithium-Ion Batteries. *International Journal of Prognostics and Health Management*, vol. 3(2) 010, pp. 1-12.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *International Journal of Advanced Manufacturing Technology*. vol. 50, pp. 297-313. doi: 10.1007/s00170-009-2482-0

Saha, B., & Goebel, K. (2011). Model Adaptation for Prognostics in a Particle Filtering Framework. *International Journal of Prognostics and Health Management*, vol. 2(1) 006, pp. 1-10Simon, D. L., Garg, S., Hunter, G. W., Guo, T. H., & Semega, K. J. (2004). Sensor needs for control and health management of intelligent aircraft engines. Technical memo. NASA/TM-2004-213202

Saxena, A., Goebel, K., Simon, D., & Eklund, Neil. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *Proceedings of International Conference on Prognostics and Health Management*. October 6-9, Denver, CO. doi: 10.1109/PHM.2008.4711414

Volponi, A. J. (2003). Foundation of gas path analysis. In Mathloudakls, K., & Sieverding, C. H., *GasTurbine Condition Monitoring and Fault Diagnosis*, (1-16). Belgium: von Karman Institute.

Wang, J., Wang, P., & Gao, R. (2013). Tool life prediction for sustainable manufacturing. *Proceedings of 11th Global Conference on Sustainable Manufacturing*, September 23-25, Berlin, Germany.

Zhu, J., Yoon, J., He, D., Qu, Y., & Bechhoefer, E. (2013). Lubrication Oil Condition Monitoring and Remaining Useful Life Prediction With Particle Filtering. *International Journal of Prognostics and Health Management*: vol. 4 020, pp. 1-15.

BIOGRAPHIES

Peng Wang is a PhD Candidate at the University of Connecticut. He received his B.S. and M.S. from Beijing University of Chemical Engineering, China in 2010 and 2013 respectively. His research interests include system estimation, health monitoring, fault diagnosis and prognosis.

Robert X. Gao received his Ph.D. from the Technical University of Berlin, Germany in 1991. He is a professor and the Pratt and Whitney Chair of Mechanical Engineering at the University of Connecticut. His research interests include signal transduction, smart structures and materials, and signal processing for diagnosis and prognosis of dynamic systems. He is a recipient of multiple awards, and a Fellow of IEEE, SME, and ASME.

Building a Data-Driven Vital Sign Indicator for an Economically Optimized Component Replacement Policy

Hyung-il Ahn[1], Ying Tat Leung[2], and Axel Hochstein[3]

[1,3] *GE Software, GE Global Research, San Ramon, CA, 94583, USA*
hyungil.ahn@ge.com
axel.hochstein@ge.com

[2] *IBM Almaden Research Center, San Jose, CA, 95120, USA*
ytl@us.ibm.com

ABSTRACT

In asset-intensive services, a well-known challenge is to maintain high availability of the physical assets while keeping the total maintenance cost low. In applications of high-value machinery such as heavy industrial equipment, a traditional approach is to perform periodic maintenance according to a runtime-based schedule. Most equipment vendors publish a maintenance schedule based on a "standard" or "average" working environment. In addition, it is a common practice that maintenance schedules from equipment vendors are highly conservative in order to reduce in-field failures which gives an adverse perception of a vendor's reputation. Therefore, such a schedule may not result in satisfactory performance as measured according to the owner's business objectives. Also, the assumption of normal operating condition may not apply in some situations. For example, stresses due to frequent overloading, continuous usage of engine at a high rate in tough environments, machine usage beyond its designed capacity can serve as good contributors to excessive wear and premature failures. In this paper we propose a novel computational framework to build a data-driven economically optimized vital sign indicator for a given component type and an economic criterion (e.g., average maintenance cost per unit runtime) by combining different sources of historical data such as total runtime hours, load carried, fuel consumed and event information from sensors. This new vital sign indicator can be viewed as a transformed time scale and used to find the optimal threshold value (or "scheduled replacement time equivalent") for a component replacement policy. Our case study was based on the collected data from 50 mining haul trucks over about 6 years in one of the largest mining service companies in the world. We present that the new vital sign indicator-based replacement policy for a critical component type largely improves on the traditional runtime-based schedule in terms of a given economic criterion, achieving a lower total maintenance cost of the enterprise.

Hyung-il Ahn et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

A traditional replacement policy for components in asset-intensive service business is often based on runtime hours-based fixed time interval ("scheduled replacement time") that the manufacturer of equipment recommends for scheduled maintenance. This is based on standard usage in an average situation assumed by the manufacturer. Most equipment vendors publish a maintenance schedule based on a "standard" or "average" working environment. In addition, it is a common practice that maintenance schedules from equipment vendors are highly conservative in order to reduce in-field failures which gives an adverse perception of a vendor's reputation. Therefore, such a schedule may not result in satisfactory performance as measured according to the owner's business objectives. Also, the assumption of normal operating condition may not apply in some situations. For example, stresses due to frequent overloading, continuous usage of engine at a high rate in tough environments, machine usage beyond its designed capacity can serve as good contributors to excessive wear and premature failures.

In asset-intensive services, a well-known challenge is to maintain high availability of the physical assets while keeping the total maintenance cost low (Jardine & Tsang, 2013). The optimization of replacement decision policy based on component failure predictions has been critical in the area of condition-based predictive asset management. One of the most popular approaches involves modeling a proportional hazard function (Cox PHM) with time-dependent covariates and a Weibull baseline hazard function

(Banjevic, Jardine, Makis, & Ennis, 2001)(Jardine, Banjevic, Montgomery, & Pak, 2008). In practice, the modeled hazard function using this approach is not guaranteed to be monotonically increasing, and thus, it often involves a complicated algorithm to compute the optimal policy (Wu & Ryan, 2011). Furthermore, a non-monotonic hazard function is not very intuitive and cannot be viewed as a new kind of time scale. Equipment managers would often like to have a time scale-like monotonically increasing measure for the component replacement policy. Then, they could use this new vital sign indicator measure exactly in the same way they used the runtime measure for replacement decisions.

In this paper we propose a novel computational framework to build a data-driven economically optimized vital sign indicator for a given component type and an economic criterion (e.g., average maintenance cost per unit runtime) by combining different sources of historical data such as total runtime hours, load carried, fuel consumed and event information from sensors. A vital sign indicator can provide a measure that contains useful information with respect to the "health" of a piece of a component or equipment, and can therefore support improved decision making in terms of maintenance planning and execution, as well as production maximization. This new vital sign indicator can be viewed as a transformed time scale and used to find the optimal threshold value (or "scheduled replacement time equivalent") for a component replacement policy. We provide an individualized maintenance plan for each component based on its real usage. Our approach involves classification and regression techniques for estimating a hazard rate and uses the "individualized" cumulative failure probability model for building a vital sign indicator.

Our case study was based on the collected data from 50 mining haul trucks over about 6 years in one of the largest mining service companies in the world. We present that the new vital sign indicator-based replacement policy for a critical component type largely improves on the traditional runtime-based schedule in terms of a given economic criterion, achieving a lower total maintenance cost of the enterprise.

2. COMPONENT REPLACEMENT POLICIES

2.1. Runtime-based Replacement Policy

Figure 1 shows an example of the failure probability density function with T^* (optimal scheduled replacement time) for a component type. Assuming that a company has run a scheduled replacement policy at T^*, at the time of collecting the component data for our analysis, the historical list of all components of this component type over a group of equipment include running components (at the time of data collection), schedule-replaced components, and failure-replaced components. In Figure 1 each circle represents a component in the list. All blue circles before T^* correspond to running components and their observed runtimes at the time of data collection. All blue circles after T^* correspond to schedule-replaced components. Note that companies in practice often do not keep the exact replacement schedule at T^*. All red circles before T^* correspond to in-field failure replacements. Note that running and scheduled replacement components are considered "right-censored" samples in survival analysis. That is, we know that the components survived at the time of data collection or scheduled replacement, but cannot tell when those components would actually fail in the future.

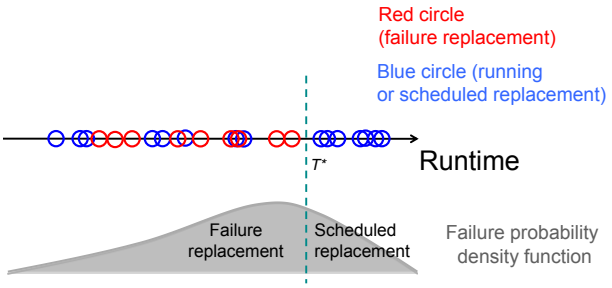


Figure 1. An example of failure probability density function with the optimal scheduled replacement time T^*

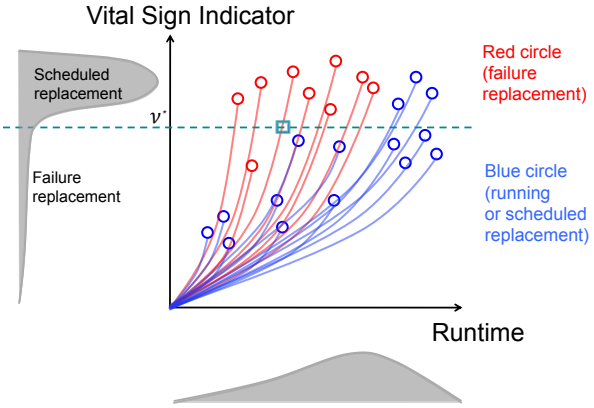


Figure 2. An example of vital sign indicator with the optimal scheduled replacement vital sign value v^*

Note in Figure 1 that the standard deviation of the failure probability density function is very large; thus, we have too many in-field failure-replaced components

2.2. Vital sign-based Replacement Policy

Now we conceptually explain the development of our new vital sign indicator model. For the historical list of all components, we also have the corresponding time-stamped logs of runtime hours (meter), total fuel consumption, total work (load) and sensor events. Imagine that for the component data and the failure probability density function shown in Figure 1, we can design a vital sign indicator

(vertical axis) in Figure 2 using some features derived from all available information. Note that the time/color of each circle in Figure 2 are exactly the same as those of the corresponding circle in Figure 1, and the color (failure replacement (red), running or scheduled replacement (blue)) of each path is based on the collected component data (i.e., the traditionally employed runtime-based replacement policy), not according to the new vital sign-based replacement policy.

Then, we propose a vital sign indicator-based scheduled replacement policy that replaces components when their vital sign value reaches a threshold value v^*. In Figure 2, the dotted line shows the threshold value. Each path in the runtime vs. vital sign indicator 2-dimensional plot corresponds to a component and shows its vital sign indicator profile over the runtime. Note that the runtime (= the value in the horizontal axis) at the intersection point between the threshold line and the path for a component indicates the actual replacement time using the policy.

Keep in mind that the failure probability density function in terms of the vital sign indicator axis depends on our model of a vital sign indicator. Intuitively, one desirable characteristic for being a good vital sign indicator is a small standard deviation in the vital sign indicator axis. This contributes to a better classification, using a constant v^*, between the failure-replaced components (above the v^* line) and the other running/schedule-replaced components (below the v^* line). In other words, if this vital sign indicator-based scheduled replacement policy had been used in the past, most of failure-replaced components in the collected data (red circles) would have been replaced on schedule (at v^*) before the actual in-field failures. However, this characteristic about the failure probability is not a sufficient condition to be a good vital sign indicator model, since the average runtime to scheduled replacements (i.e., the average of actual runtimes from intersection points at v^*) and the average runtime to failure replacements should also be large values. For this reason, we should look into the shape of vital sign paths in the runtime vs. vital sign indicator 2-dimensional plot. We will explain it using economic optimization equations below in more detail.

3. ECONOMIC OPTIMIZATION

3.1. Runtime-based Replacement Policy

Let $F(t)$ be the cumulative failure probability function at runtime t (= $\Pr(T \le t)$ where T is a random variable denoting the runtime at failure), $S(t) = 1 - F(t)$ be the survival probability function at t. When we deal with the dataset from real industry practice, it is very likely that there is no failure data after the scheduled replacement time the company has employed during the period of the dataset. Therefore, we would not make a good estimate on the exact shape of the function over the time after the current scheduled replacement time. However, in this paper we assume that the survival probability function can be estimated using a parametric Weibull fit (Fox, 2002) to the runtime and failure data.

For our economic optimization analysis, we are provided the economic and logistic parameters including

C_f = in-field failure replacement cost, which includes the part and labor cost to replace the component, the retrieval cost of equipment from the field, and lost revenue due to blocking other equipment when it fails in the field (called "circuit break"),

C_p = scheduled replacement cost, which includes the part and labor cost to replace the component,

c_d = cost per unit downtime of the equipment, including lost revenue that could have been contributed by that piece of equipment,

DT_f = down time due to an in-field failure,

DT_p = down time due to a scheduled replacement.

In general, in-field failure replacement cost and downtime are greater than scheduled replacement cost and downtime, respectively ($C_f > C_p$, $DT_f > DT_p$).

Denote by t_p the scheduled replacement time for the policy, which is our optimization target. With this scheduled replacement policy, the mean time to failure replacement that happens before t_p is denoted by t_f and estimated as:

$$t_f = \frac{1}{F(t_p)} \int_0^{t_p} t f(t) dt = t_p - \frac{\int_0^{t_p} F(t) dt}{F(t_p)}$$

A new component lifetime cycle starts at the installation time of a component. The component may be replaced due to an in-field failure or a scheduled replacement finishing its lifetime cycle.

For a runtime-based replacement policy, we choose t_p to minimize the *average maintenance cost per unit runtime*.

average total time per cycle

$$= (t_f + DT_f)F(t_p) + (t_p + DT_p)(1 - F(t_p))$$

average run time per cycle $= t_f F(t_p) + t_p(1 - F(t_p))$

average maintenance cost per unit runtime

$$= \frac{average\ maintenance\ cost\ per\ cycle}{average\ runtime\ per\ cycle}$$

$$= \frac{(average\ failure\ replacement\ cost\ per\ cycle + average\ scheduled\ replacement\ cost\ per\ cycle)}{average\ runtime\ per\ cycle}$$

$$= \frac{(C_f + c_d DT_f)F(t_p) + (C_p + c_d DT_p)(1 - F(t_p))}{t_f F(t_p) + t_p(1 - F(t_p))}$$

$$= (C_f - C_p + c_d(DT_f - DT_p))\frac{F(t_p)}{X} + (C_p + c_d DT_p)\frac{1}{X}$$

where X = *average run time per cycle*

$$= t_f F(t_p) + t_p(1 - F(t_p))$$

As t_p (= the scheduled replacement time) is set to a higher value, there is more chance of in-field failure replacements, that is, $F(t_p)$ (= the total probability of in-field failure replacements) becomes larger (See Figure 3).

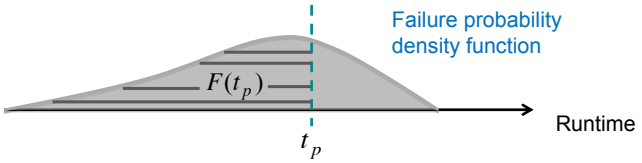


Figure 3. The trade-off between the average runtime per cycle and $F(t_p)$ (= total in-field failure probability)

Since $DT_f > DT_p$ and $C_f > C_p$ in general, the optimization goal of minimizing the average maintenance cost per unit runtime is achieved by increasing *average runtime per cycle* ($X = t_f F(t_p) + t_p(1 - F(t_p))$) and decreasing in-field failure probability per cycle $F(t_p)$. Note that there is a trade-off between decreasing $F(t_p)$ and increasing the *average runtime per cycle*. In general, decreasing $F(t_p)$ that would involve fewer failure replacements can be obtained by decreasing t_p, but this then reduces the *average run time per cycle*. Note that $t_f < t_p$ in general. Also, note that as $F(t_p)$ becomes smaller, t_p becomes more weighted in the estimate of average run time per cycle. Given $F(t), C_f, C_p, c_d, DT_f$ and DT_p, the average maintenance cost per unit runtime is a function of t_p, which is denoted as g.

$$g(t_p) = \frac{(C_f + c_d DT_f)F(t_p) + (C_p + c_d DT_p)(1 - F(t_p))}{t_f F(t_p) + t_p(1 - F(t_p))}$$

It is important to note that the cumulative failure probability function $F(t)$ is fixed and can be estimated using the failure data for the component type we analyze. Note also that t_f depends on $F(t)$. Then, the optimized time threshold for the scheduled replacement policy is $t_p^* = \arg\max_{t_p} g(t_p)$.

3.2. Vital Sign-based Replacement Policy

Let v be vital sign indicator. $\hat{F}(v)$ be the cumulative failure probability function at vital sign v (= $\Pr(V \le v)$ where V is a random variable denoting the vital sign at failure), $\hat{S}(v) = 1 - \hat{F}(v)$ be the survival probability function at v. Note that we estimate this survival probability function by a local regression (loess) on the Kaplan-Meir (KM) estimate (Therneau, 2000) using the vital sign and failure data.

Denote by v_p the vital sign threshold value for scheduled replacements for the vital sign-based scheduled replacement policy, which is our optimization target. Then, $\hat{F}(v_p)$ is the total expected probability of failure replacements, and $1 - \hat{F}(v_p)$ is the total expected probability of scheduled replacements. With this scheduled replacement policy, the expected time to scheduled replacement at v_p is denoted by $\hat{t}_p$. Also, the expected time to failure replacement is denoted by $\hat{t}_f$. In this paper we estimate $\hat{t}_p$ and $\hat{t}_f$ under reasonable assumptions.

Let $Comp[v \ge v_p]$ denote the set of all components whose vital sign value reaches v_p in the dataset, whereas $Comp[v < v_p]$ denotes the set of all components whose vital sign value $v < v_p$ for all time t in the dataset.

Let $P[v \ge v_p]$ denote the *actual* ratio of the number of components in $Comp[v \ge v_p]$ to the total number of components *in the dataset*. The actual ratio $P[v \ge v_p]$ is equal to or smaller than $1 - \hat{F}(v_p)$ (= total expected probability of scheduled replacement), since the total expected probability takes right-censored components (running at the time of data collection) into account. There are running components that would fail with $v > v_p$. We assume that those components contribute to scheduled replacements corresponding to the difference between the expected probability and the actual ratio (= $1 - \hat{F}(v_p) - P[v \ge v_p]$) and that they are schedule-replaced at v_p with the cumulative probability function of the replacement time, $\hat{F}_{v<v_p}(t) = 1 - \hat{S}_{v<v_p}(t)$ where $\hat{S}_{v<v_p}(t)$ is the survival probability function estimated using a Weibull fit to the runtime and failure data of $Comp[v < v_p]$. In other words, we assume that $\hat{F}_{v<v_p}(t)$ estimated using $Comp[v < v_p]$ is uniformly applied to all the range of $v < v_p$. Thus, the mean scheduled replacement time over those components corresponding to $1 - \hat{F}(v_p) - P[v \ge v_p]$ is the same as the mean failure time over $Comp[v < v_p]$, which is denoted by r and estimated as $r = \int_0^\infty \hat{S}_{v<v_p}(t)dt$. Thus,

$\hat{t}_f$ = expected time to failure replacement $= \int_0^\infty \hat{S}_{v<v_p}(t)dt$.

$\hat{t}_p$ = expected time to scheduled replacement

$= \{ P[v \geq v_p]\, E[t|v = v_p \text{ for } Comp[v \geq v_p]] +$

$(1-\hat{F}(v_p) - P[v \geq v_p])\, r \} / (1-\hat{F}(v_p))$.

Note that $E[t|v = v_p \text{ for } Comp[v \geq v_p]]$ is the average of scheduled replacement times at $v = v_p$ over $Comp[v \geq v_p]$.

Alternatively, we may assume that components in $Comp[v < v_p]$ that would fail after t_c contribute to scheduled replacements for the difference $(= 1-\hat{F}(v_p) - P[v \geq v_p])$, whereas components in $Comp[v < v_p]$ that would fail before t_c are failure-replaced. Also, we can estimate t_c from the constraint $\hat{F}(v_p) = \hat{F}_{v<v_p}(t_c)\ (1 - P[v \geq v_p])$. That is, the total expected probability of failure replacements over all components $(= \hat{F}(v_p))$ should be the same as the actual ratio of the number of components in $Comp[v < v_p]$ to the number of total components in the dataset $(= 1 - P[v \geq v_p])$ multiplied by the total expected probability of failure replacements before t_c over $Comp[v < v_p]$ $(= \hat{F}_{v<v_p}(t_c))$. Thus,

$\hat{t}_f$ = expected time to failure replacement

$$= t_c - \frac{\int_0^{t_c} \hat{F}_{v<v_p}(t)dt}{\hat{F}_{v<v_p}(t_c)}.$$

Then, the mean scheduled replacement time over those components corresponding to $1-\hat{F}(v_p) - P[v \geq v_p]$ is denoted by r and estimated as

$$r = \{ \int_0^\infty \hat{S}_{v<v_p}(t)dt - \hat{t}_f\ \hat{F}_{v<v_p}(t_c) \} / (1-\hat{F}_{v<v_p}(t_c)).$$

$\hat{t}_p$ = expected time to scheduled replacement

$= \{ P[v \geq v_p]\, E[t|v = v_p \text{ for } Comp[v \geq v_p]] +$

$(1-\hat{F}(v_p) - P[v \geq v_p])\, r \} / (1-\hat{F}(v_p))$.

For this vital sign-based replacement policy, we choose v_p to minimize the *average maintenance cost per unit runtime*.

Average maintenance cost per unit runtime

$$= \frac{\textit{average maintenance cost per cycle}}{\textit{average runtime per cycle}}$$

$$= \frac{(C_f + c_d DT_f)\hat{F}(v_p) + (C_p + c_d DT_p)(1-\hat{F}(v_p))}{\hat{t}_f \hat{F}(v_p) + \hat{t}_p (1-\hat{F}(v_p))}$$

$$= (C_f - C_p + c_d(DT_f - DT_p))\frac{\hat{F}(v_p)}{\hat{X}} + (C_p + c_d DT_p)\frac{1}{\hat{X}}$$

where $\hat{X}$ = *average run time per cycle*

$$= \hat{t}_f \hat{F}(v_p) + \hat{t}_p (1-\hat{F}(v_p))$$

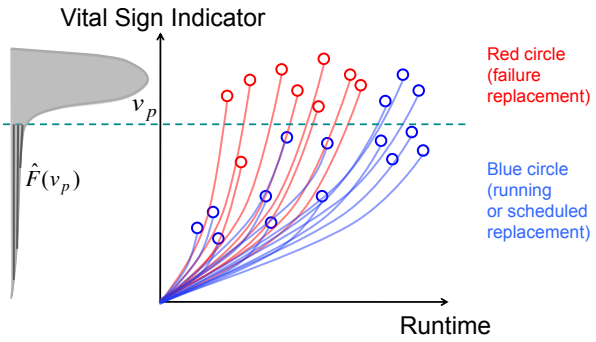


Figure 4. Vital-sign indicator functions steeply increasing around v_p: no strong trade-off between the average runtime per cycle and $\hat{F}(v_p)$ (= total in-field failure probability)

As in the analysis of the runtime-based policy, the optimization goal of minimizing *average maintenance cost per unit work* is achieved by increasing *average run time per cycle* $(= \hat{t}_f \hat{F}(v_p) + \hat{t}_p (1-\hat{F}(v_p)))$ and decreasing in-field failure probability per cycle $\hat{F}(v_p)$. However, in contrast to the runtime-based policy, with vital-sign indicator functions steeply increasing around v_p, there is no strong trade-off between decreasing $\hat{F}(v_p)$ and increasing the *average run time per cycle*. In other words, decreasing $\hat{F}(v_p)$ that would involve fewer failure replacements can be obtained by decreasing v_p but this does not necessarily lead to a large decrease of $\hat{t}_p$ (= the average of scheduled replacement times at v_p) when the vital-sign indicator functions are steeply increasing around v_p (compared with slowly increasing shaped functions). More importantly, considering the definitions of $\hat{t}_p$ (involving the term $[t|v = v_p \text{ for } Comp[v \geq v_p]]$) and $\hat{t}_f$ (involving $\hat{S}_{v<v_p}(t)$ or $\hat{F}_{v<v_p}(t)$), if decreasing v_p would allow failures that happen *later in time* to be schedule-replaced, this would tend to increase both $\hat{t}_p$ and $\hat{t}_f$, as well as decreasing $\hat{F}(v_p)$; thus, this helps the optimization

goal. Also, if decreasing v_p would allow failures that happen *earlier in time* to be schedule-replaced, this would tend to decrease $\hat{t}_p$ but still tend to increase $\hat{t}_f$ and decrease $\hat{F}(v_p)$. Note that in contrast to the runtime-based policy, $\hat{t}_f$ is not necessarily smaller than than $\hat{t}_p$ for a vital sign-based policy. That is, decreasing $\hat{t}_p$ does not lead to decreasing $\hat{t}_f$. The values of $\hat{t}_p$ and $\hat{t}_f$ at the optimization of v_p rely on the complete distribution and paths in the runtime vs. vital sign indicator 2-dimensional plot.

It is critical to note that the shape of cumulative failure probability function $\hat{F}(v')$ for any candidate threshold v' can be changed according to our modeling parameters to design a vital sign indicator. Note also that $\hat{t}_p$ and $\hat{t}_f$ for any candidate threshold v' (i.e., functions of v') depend on the designed vital sign indicator.

Given $C_f, C_p, c_d, DT_f, DT_p$, $\hat{F}(v')$, $\hat{t}_p(v')$ and $\hat{t}_f(v')$ for a designed vital sign indicator, the average maintenance cost per unit runtime is a function of v_p, which is denoted as $\hat{g}$.

$$\hat{g}(v_p \mid \hat{F}(v'),\ \hat{t}_p(v'),\ \hat{t}_f(v')) =$$

$$\frac{(C_f + c_d DT_f)\hat{F}(v_p) + (C_p + c_d DT_p)(1 - \hat{F}(v_p))}{\hat{t}_f(v_p)\hat{F}(v_p) + \hat{t}_p(v_p)(1 - \hat{F}(v_p))}$$

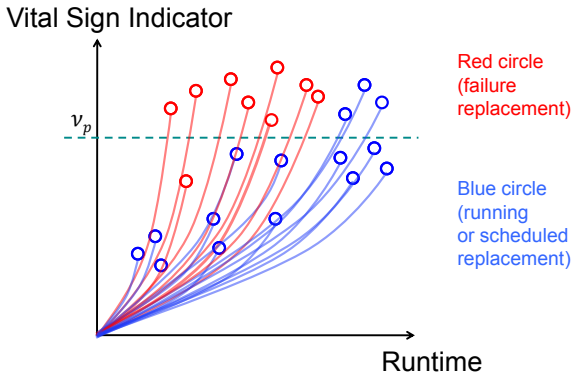
Thus, the value of $\hat{g}$ at v_p is determined by our design of the vital sign indicator, which is what the paths of vital sign over time look like.

Then, the optimized vital sign threshold value for the scheduled replacement policy using this vital sign indicator is $v_p^* = \arg\max_{v_p} \hat{g}(v_p \mid \hat{F}(v'), \hat{t}_p(v'),\ \hat{t}_f(v'))$.

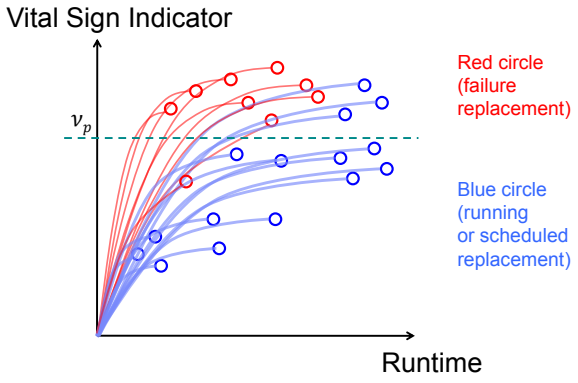
We compare the runtime-based component replacement policy with the new designed vital sign-based replacement policy in terms of the average maintenance cost per unit runtime. That is, we compare $g(t_p^*)$ with $\hat{g}(v_p^* \mid \hat{F}(v'), \hat{t}_p(v'),\ \hat{t}_f(v'))$.

If $\hat{g}(v_p^* \mid \hat{F}(v'), \hat{t}_p(v'),\ \hat{t}_f(v')) > g(t_p^*)$, this means that the designed vital-sign based replacement policy is more beneficial in terms of the economic criterion.

4. BUILDING A VITAL SIGN INDICATOR BASED ON CLASSIFICATION AND REGRESSION



(a) Convex-shaped vital sign indicator model



(b) Concave-shaped vital sign indicator model

Figure 5. Comparing convex-shaped and concave-shaped vital sign indicator models

In Figure 5(a) and (b), we compare two hypothetical vital sign indicator models (convex-shaped and concave-shaped) when the failure probability density functions in the vital sign indicator axis are the same, although this would hardly happen in practice. For the same vital sign threshold value v_p, the convex shape in Figure 5 (a) would have a greater average runtime to scheduled replacement ($\hat{t}_p$ = the average of runtimes from all intersection points) than the concave shape in Figure 5 (b). The convex paths would predict the upcoming failures near the actual failure times, whereas the concave paths would predict the upcoming failures too early. The concave paths would have a smaller average runtime due to too early replacements. Thus, in general, the convex-shaped vital sign indicator model would be more desirable than the concave-shaped one. This is also why we should look into the complete vital sign paths, not just examining the shape of failure probability density function or $\hat{F}(v_p)$.

Before explaining our vital sign indicator model, we first introduce the notion of "*individualized* cumulative failure

probability function". For each individual component, let us consider a hypothetical population of components that share the same history of covariates as that component has. Then, we can define a cumulative distribution function of the failure time for the population. We call it the individualized cumulative failure probability function for the component. In addition, the individualized cumulative failure probability function $F_j(t)$ of component j has the following relationship with the individualized cumulative hazard function $H_j(t)$:

$F_j(t) = 1 - S_j(t) = 1 - \exp(-H_j(t))$ where $S_j(t)$ is the individualized survival probability function.

In this paper we model the vital sign indicator using the individualized cumulative failure probability function. That is, the vital sign indicator for a component is the same as its individualized cumulative failure probability over runtime.

In the runtime-based policy we select the best scheduled replacement time so that the cumulative failure probability $F(t_p)$ optimizes the economic criterion. In contrast, in the vital sign-based policy for scheduled replacements, we apply a selected vital sign threshold value to the individualized cumulative failure probability functions $F_j(t)$ of components. This is the same as applying a common threshold to the individualized cumulative hazard functions $H_j(t)$. Note that this individualization in cumulative failure probability (or cumulative hazard) is critical to allow each component to have its own transformed time scale for the replacement policy.

The individualized cumulative hazard $H_j(t)$ assesses the total amount of accumulated risk that the component j has faced from the beginning of time until the present, while the (instantaneous) hazard rate assesses the risk that a component which has not yet had the failure so will experience it within a unit of runtime (Singer & Willett, 2003). Compared to using the hazard rate in designing a scheduled replacement policy, applying the individualized cumulative hazard $H_j(t)$ has some advantages. First, in contrast to the hazard rate, the individualized cumulative hazard may capture the accumulated wear and tear over the component runtime. Second, the individualized cumulative hazard is always increasing, whereas the hazard rate may be fluctuating up and down over the runtime. Note that the characteristic of monotonically increasing is necessary because the vital sign indicator is conceptualized as a transformed time scale. In addition, people usually think that the accumulated wear and tear is always increasing over the runtime, that is, the quality of a component becomes worse with runtime.

Considering that our dataset includes daily-interval samples, we define the daily hazard $h_j(d)$ on date d for component j by the total hazard during the daily runtime. That is, daily hazard = hazard rate × daily runtime. Then, we can estimate the individualized cumulative hazard by summing up all daily hazards until the present time t: $H_j(t) = \sum_{\text{all } d \text{ in } \{d:Meter(j,d)\le t\}} h_j(d)$ where $Meter(j,d)$ is the accumulated runtime hours over days up to and including date d.

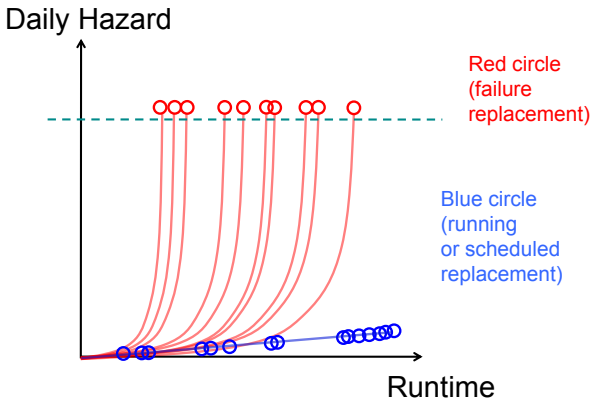


Figure 6. An example of the "designed" daily hazard as a regression target variable

It is important to note that the "estimated" daily hazard depends on our selection of covariates and the model. Also, daily hazard estimates from a desirable model would predict its failure near the date of actual failure time. Wrong predictions or too early predictions of failures would lead to the reduction of average runtime. Thus, it will be better to find the covariates and model that enable the daily hazard estimates to be convex-shaped and very close to the maximum value (= 1) near the date of actual failure time (e.g., Figure 6). In practice, however, we do not require the daily hazard estimates to be necessarily convex-shaped, because it may not be possible with our selected features and modeling choice. We only want the individualized cumulative hazards to satisfy some desired characteristics (monotonically increasing, high values of $\hat{t}_p$ and $\hat{t}_f$, high vital sign values on the failure times) for the economic criterion. Thus, we set up our problem of designing a vital sign indicator model as a regression task where the regression target variable is the "designed" daily hazard $\tilde{h}_j(d)$ we specify on any date d for component j as follows:

- If the component was failure-replaced, $\tilde{h}_j(d) = (Meter(j,d)/Meter(j,T_F(j)))^{\alpha}$ where $Meter(j,d)$ is the total runtime hours up to and including date d, $T_F(j)$ is the finally observed date (or the replaced date), and $\alpha \ge 1$.

- If the component was schedule-replaced or actively running, $\tilde{h}_j(d) = \beta(Meter(j,d)/M_{max})^{\alpha}$ where $M_{max} = max_i[Meter(i,T_F(i))]$ = the maximum total runtime hours over all components in the dataset, and β ($\ll 1$) is a small positive number close to 0 (e.g., $\beta = 0.1$).

That is, shown as in Figure 6, the first equation satisfies the condition that failure-replaced components have the maximum value (= 1) near the date of actual failure time. Also, the second equation allows the running/schedule-replaced components to have low values over their runtimes.

We build vital sign indicator models by performing regression tasks with differently designed daily hazard setups (different α and β values), and find the best vital sign indicator model in terms of the economic optimization criterion estimate by leave-one-component-out cross-validations. We will describe it below in detail.

Provided that we have the list of past replaced components (failure or scheduled replacements) and current running ones for a component type over a group of equipment as well as the corresponding time-stamped logs of runtime hours (meter), total fuel consumption, total work (load) and sensor events, we propose a framework of building a vital sign indicator for the component type using regression.

Suppose that there are totally J components that were past replaced or are actively running for the target component type. For component j (=1, …, J), the start date of service is $T_S(j)$, and the final date of observation is $T_F(j)$. Note that the final date of observation is defined as the replaced date for past components or the last observed date for running components. For this task, the overall dataset includes all points $x(j,d)$ over component j (=1, …, J) and date d (=$T_S(j)$, …,$T_F(j)$).

Input data:
From the start date of service of component j,

- *Meter(j,d)* = accumulated runtime hours over days up to and including date d
- *Fuel(j,d)* = accumulated fuel consumption over days up to and including date d
- *Load(j,d)* = accumulated number of loads (total work) over days up to and including date d
- *EventCount(j,d)* = accumulated number of relevant sensor events for the target component type over days up to and including date d

Note that $Meter(j, T_S(j)) = 0$, $Fuel(j, T_S(j)) = 0$, $Load(j, T_S(j)) = 0$, and $EventCount(j, T_S(j)) = 0$. Here we assume that the relevant sensor event types for the component type are selected using the significance test in a univariate Cox proportional hazard model for each event type (Hastie, Tibshirani, Friedman, & Franklin, 2005)(Bair, Hastie, Paul, & Tibshirani, 2006). But other techniques including frequent sequence mining (Zaki, 2001) on component failure and event data can be exploited for the same purpose.

Given the parameters such as

N_{smooth} = positive integer for a smoothing filter,

N_{fuel} = positive real threshold value for counting the number of dates with high daily fuel rate,

N_{load} = positive real threshold value for counting the number of dates with high daily load rate,

we compute intermediate variables as follows. Note that these intermediate variables are used to calculate features. Also, the purpose of N_{fuel} and N_{load} is to count outliers. Although we present this simple rule-based outlier detection here, our framework allows other sophisticated anomaly detection algorithms to be applied for more effective feature generation.

Intermediate variables:

- *DailyMeter*(j,d) = daily meter hours on date d
  $= Meter(j,d) – Meter(j,d-1)$
- *DailyFuel*(j,d) = daily fuel consumption on date d
  $= Fuel(j,d) – Fuel(j,d-1)$
- *DailyLoad*(j,d) = daily number of loads on date d
  $= Load(j,d) – Load(j,d-1)$
- *SmoothedDailyMeter(j,d)* = average daily meter hours over past N_{smooth} days on date d
- *SmoothedDailyFuel(j,d)* = average daily fuel consumption over past N_{smooth} days on date d
- *SmoothedDailyLoad(j,d)* = average number of loads over past N_{smooth} days on date d
- *DailyFuelRate(j,d) = SmoothedDailyFuel(j,d) / SmoothedDailyMeter(j,d)*
- *DailyLoadRate(j,d) = SmoothedDailyLoad(j,d) / SmoothedDailyMeter(j,d)*
- *HighFuelRateCount(j,d)* = accumulated count of days in which the daily fuel rate > N_{fuel} over days up to and including date d
- *HighLoadRateCount(j,d)* = accumulated count of days in which the daily load rate > N_{load} over days up to and including date d

Before doing the regression task, we perform a classification task to estimate the probability of having the component failure within next M runtime hours from each date. This estimated failure probability can be used as a key predictor variable in the later regression task. We observed that this failure probability improved fitting to the designed daily hazard in the regression task.

For the classification task, we now explain how to compute features and assign labels to model the predicted failure probability.

Features for the classification task:

- *HighFuelRateCountPerMeter(j,d) = HighFuelRateCount(j,d) / Meter(j,d)*
- *HighLoadRateCountPerMeter(j,d) = HighLoadRateCount(j,d) / Meter(j,d)*
- *TotalFuelRate(j,d) = Fuel(j,d) / Meter(j,d)*
- *TotalLoadRate(j,d) = Load(j,d) / Meter(j,d)*
- *TotalEventRate(j,d) = EventCount(j,d) / Meter(j,d)*

Label assignment for the classification task:
We assign the classification label *L(j,d)* to each point *x(j,d)* that corresponds to date *d* for component *j*. Note that *x(j,d)* is a multi-dimensional vector of classification features. Among all historical data of component replacements, there are two types of replacement on the final date of observation: scheduled replacement and in-field failure replacement. The goal of the classification task is to estimate the failure probability within the next *M* runtime hours from each date *d*. With binary classification labels of *Failure* and *No Failure* classes,

- For a point *x(j,d)* on a failure-replaced component *j*, when *Meter(j, d)* is within *M* meter hours of the failure replacement (that is, *Meter(j,d) > Meter(j, T_F(j)) - M*), classification label *L(j,d)* is assigned *Failure* class. Otherwise, classification label *L(j,d)* is assigned *No Failure* class.
- For any point *x(j,d)* on a schedule-replaced component *j*, classification label *L(j,d)* is assigned *No Failure* class.
- For any point *x(j,d)* on running component *j*, classification label *L(j,d)* is assigned *No Failure* class.

To measure the performance of our model, we propose and use leave-one-component-out cross validation. That is, for each run corresponding to a component *j* (= 1,…, *J*), we split the overall dataset into the *test dataset* of all points from component *j* and the *training dataset* of all points from all *J-1* remaining components *k* (≠ *j*), build a vital sign indicator model based on the training dataset only and compute the vital sign indicator values on all points in the test dataset. In more detail, we have *J* runs in total, and in each run corresponding to a component *j* we perform the steps below.

Initial Parameters: α and β (designing daily hazards), N_{smooth}, N_{fuel}, N_{load} (computing features), *M* (modeling failure probability)

Step 1. Divide the overall dataset into the test dataset of all points from one component *j* and the training dataset of all points from remaining components.

Step 2. Using only the *training dataset*, perform the classification to build a binary classifier (e.g., applying Support Vector Classification (Cristianini & Shawe-Taylor, 2000)) to compute the failure probability $P_{failure}(j, d)$ (= probability of being *Failure* class) on each point. This estimated probability can be viewed as the failure probability within the next *M* runtime hours from date *d*.

Step 3. Design the target variable for the regression task. This regression target variable $\tilde{h}_k(d)$ for any component *k* (≠ *j*) in the training dataset should have the desired characteristic of the daily hazard such as being monotonically increasing, convex-shaped, and the maximum value on failure.

Step 4. Using only the *training dataset*, build the regression model (e.g., applying Support Vector Regression (Scholkopf & Smola, 2002) to target daily hazard $\tilde{h}_k(d)$ with feature variables such as *Meter(k,d)*, *Fuel(k,d)*, *Load(k, d)*, *EventCount(k,d)* and $P_{failure}(j, d)$.

Step 5. Apply the built regression model to obtain the estimated daily hazard $h_j(d)$ for each point *x(j,d)* on component *j* in the *testing dataset*.

Step 6. Compute the individualized cumulative hazard on component *j*, $H_j(t) = \sum_{\text{all } d \text{ in } \{d:Meter(j,d)\le t\}} h_j(d)$.

Step 7. Compute the individualized cumulative failure probability on component *j*, $F_j(t) = 1 - \exp(-H_j(t))$.

After all *J* runs in leave-one-component-out cross validations, we can obtain the vital sign indicator values over all components. Given these values, we perform an optimization task to obtain the optimal threshold value for the replacement policy in terms of the economic optimization criterion such as the average maintenance cost per unit runtime. Note that in a threshold-based replacement policy, a component should be replaced when the vital sign indicator value reaches a threshold value. Optionally, we may use this estimated optimal threshold value to normalize the vital sign indicator. Then, a component should be replaced when its vital sign is 100% of wear.

In general, the parameter selections (α, β, N_{smooth}, N_{fuel}, N_{load}, *M*) influence the ultimate model. Thus, we need to find the optimal parameters to obtain the best vital sign indicator model in terms of our optimization criterion.

5. CASE STUDY

Our proposed framework of building the vital sign indicator and optimizing the economical profit was tested with one of the largest mining service companies in the world. The collected data includes the logs of daily fuel consumption, daily number of loads moved, daily meter hours, sensor event data, and component replacement history on 50 mining haul trucks over the period from January 1st 2007 to November 11th 2012. Each truck is equipped with a set of sensors triggering events on a variety of vital machine conditions. Note that the estimated overall cost of downtime for one of these haul trucks amounts to about 1.5 million USD per day. Therefore, the financial impact of reducing the downtime is very large. This is because not only is the scheduled maintenance cost high, the total cost due to unscheduled in-field failure is even higher. When one piece of equipment breaks down, in addition to stopping its own production, it may block other equipment from producing. The goal of our vital sign indicator is to optimize the tradeoff between scheduled replacement cost and unscheduled failure cost, to achieve a lower total maintenance cost of the enterprise.

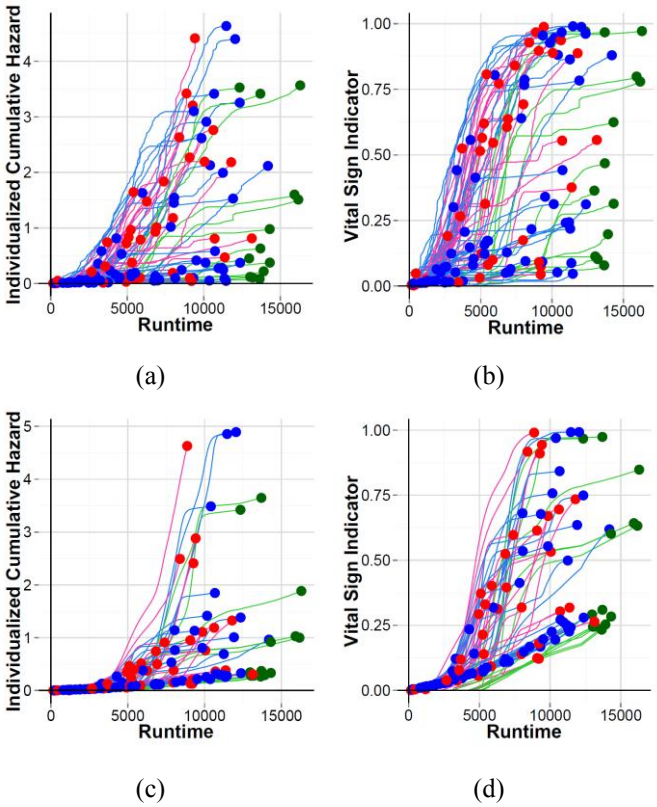


Figure 7. The individualized cumulative hazard and the vital sign indicator, (a) and (b) from SVC+SVR model, (c) and (d) from SVC+Cox model. Red = Failure replacements, Green = Scheduled replacements, and Blue = Running at the time of data collection

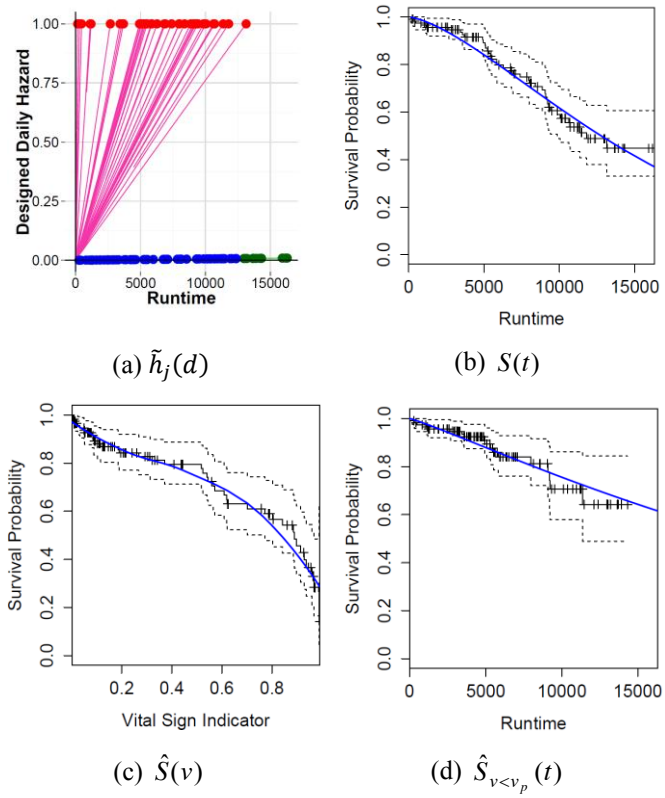


Figure 8. (a) Designed daily hazard ($\alpha = 1, \beta = 0.1$), (b) Survival probability in runtime (KM, Weibull), (c) Survival probability in vital sign indicator (KM, loess), (d) Survival probability for $Comp[v < v_p]$ (KM, loess)

In this section we present our application and results focused on one specific component type (called "X1"). To use our framework explained in the steps above, we should choose a pair of classification and regression algorithms. In general we can apply any algorithms for this purpose, but here we mainly present our results using Support Vector Classification (SVC) and Support Vector Regression (SVR). We found out that these algorithms using kernel tricks worked better than other basic algorithms including linear/quadratic discriminant analysis, generalized linear models and Cox PH regression. Also, we compared vital sign indicator models obtained using different parameter settings of α, β (designing daily hazards), N_{smooth}, N_{fuel}, N_{load} (computing features) and M (modeling failure probability) in terms of our optimization criterion. Here we show the result with the RBF kernel and the best setting of $\alpha = 1, \beta = 0.1$, $N_{smooth} = 60$, $N_{fuel} = 190$, $N_{load} = 3.0$, $M = 4890$ in our application.

Table 1. Comparison between the traditional runtime-based policy and the vital sign indicator-based policy

	Runtime-based policy	Vital sign-based policy
Threshold	$t_p = 16500$	$v_p = 0.50$
Total failure probability	$F(t_p) = 0.63$	$\hat{F}(v_p) = 0.21$
Expected time to scheduled replacement	$t_p = 16500$	$\hat{t}_p = 14848$
Expected time to failure replacement	$t_f = 8708$	$\hat{t}_f = 7311$
Avg runtime per cycle	11592	13201
Avg failure replacement cost per unit runtime	\$30.6	\$9.3
Avg scheduled replacement cost per unit runtime	\$15.6	\$27.9
Avg maintenance cost per unit runtime	\$45.6	\$37.2

The economic and logistic parameters for the target component type are as follows: C_f = failure replacement cost = \$443600, C_p = scheduled replacement cost = \$374400, c_d = cost per unit downtime of the equipment = \$2000, DT_f= down time due to an in-field failure = 64.8 hrs, DT_p = down time due to a scheduled replacement = 48 hrs. Note that $(C_f + c_d DT_f)/(C_p + c_d DT_p) = 1.22$.

Figure 7(a) and (b) show the individualized cumulative hazard and the vital sign indicator, respectively, for the model based on SVC and SVR. In the figures, each line corresponds to a component. The color of the line and corresponding end point indicates whether the component had a failure replacement at the end (red), were running at the time of data collection (blue, right-censored) or had a scheduled replacement at the end (green, right-censored).

Figure 8(a) shows the designed daily hazard. The optimized vital sign threshold was 0.50. Based on two different approaches explained to estimate $\hat{t}_p$ and $\hat{t}_f$, we obtained almost similar values of the criterion (\$37.1 and \$37.2). Figure 8(b),(c) and (d) show survival probabilities such as $S(t)$, $\hat{S}(v)$ and $\hat{S}_{v<v_p}(t)$. Considering that the cumulative failure probability corresponds to 1 − survival probability (that is, $F(t) = 1 - S(t)$, $\hat{F}(v) = 1 - \hat{S}(v)$), note that $F(t_p) = 0.63 > \hat{F}(v_p) = 0.21$. This significant reduction in total expected failure probability is a necessary condition for being a good vital sign indicator. Also, comparing $\hat{S}_{v<v_p}(t)$ and $S(t)$ in Figure 8(b) and (d), we find that the expected lifetime of $Comp[v < v_p]$ alone is significantly longer than that of all components in the dataset.

Table 1 compares the runtime-based and vital-sign based replacement policy in terms of the average maintenance cost per unit runtime. There is about 20% cost reduction with the vital-sign based policy, compared to the runtime-based policy. The new vital-sign based policy with vital sign threshold = 0.5 has some false failure predictions so involves higher average scheduled replacement cost per unit runtime than the runtime-based policy (\$27.9 > \$15.6), but the vital-sign based policy has significantly smaller average failure replacement cost per unit runtime (\$9.3 << \$30.6) and thus, overall it is better than the runtime-based policy.

We tested Cox PH regression in combination with SVC in our framework. In fact we compared several Cox PH regression models using differently selected features as time-dependent covariates. Then, we observed that the Cox PH regression simply using the SVC-estimated failure probability as the only one time-dependent covariate worked best among them. Figure 7(c) and (d) show the individualized cumulative hazard and the vital sign indicator from this model. But, this still performed a bit worse (\$38.0) than the SVR-based model (\$37.2). Note that while Cox PH regression considers only the covariate values at sampled failure times (i.e., maximizing the partial likelihood), SVR can consider covariate values at all times (i.e., maximizing the fit to the complete paths of the designed target daily hazards).

6. CONCLUSION AND DISCUSSION

We compared our vital sign indicator-based policy with a traditional runtime-based policy in terms of the average maintenance cost per unit runtime. When the failure replacement cost of a component is extremely high, it is critical to reduce the total number of in-field failures by following the recommended option for decreasing the total expected probability of failures. We modeled our vital sign indicator based on "individualized" cumulative failure probability function for each component. This new indicator as a transformed time scale allows us to have an individualized maintenance plan for each component based on its real usage. Our case study demonstrates that the new vital sign indicator-based replacement policy can obtain greater economic value in terms of the average maintenance cost per unit runtime.

Future work will include a remaining useful lifetime (RUL) model based on this vital-sign indicator. This will involve the estimation of paths in the runtime vs. vital sign indicator 2-dimensional plot. Another future direction is to incorporate a constrained regression to make vital sign indicators suitably convex-shaped, eventually leading to lower optimal costs.

REFERENCES

Bair, E., Hastie, T., Paul, D., & Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association*, *101*(473).

Banjevic, D., Jardine, A., Makis, V., & Ennis, M. (2001). A control-limit policy and software for condition-based maintenance optimization. *INFOR-OTTAWA-*, *39*(1), 32–50.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Fox, J. (2002). Cox proportional-hazards regression for survival data. *See Also*.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Jardine, A., Banjevic, D., Montgomery, N., & Pak, A. (2008). Repairable system reliability: Recent developments in CBM optimization. *International Journal of Performability Engineering*, *4*(3), 205.

Jardine, A. K., & Tsang, A. H. (2013). *Maintenance, replacement, and reliability: theory and applications*. CRC press.

Scholkopf, B., & Smola, A. (2002). *Learning with kernels*. MIT press Cambridge.

Singer, J. D., & Willett, J. B. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford university press.

Therneau, T. M. (2000). *Modeling survival data: extending the Cox model*. Springer.

Wu, X., & Ryan, S. M. (2011). Optimal replacement in the proportional hazards model with semi-markovian covariate process and continuous monitoring. *Reliability, IEEE Transactions on*, *60*(3), 580–589.

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, *42*(1-2), 31–60.

Aircraft Hydraulic System Leakage Detection and Servicing Recommendations Method

Wlamir O. L. Vianna[1], João Pedro P. Malere[2]

[1,2]*EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil*

wlamir.vianna@embraer.com.br
joao.malere@embraer.com.br

ABSTRACT

Aircraft hydraulic systems are composed of several components connected and distributed along the aircraft. Monitoring leakage of these components are time consuming tasks, and often cover only some parts of the system. The objective of this work is to present a method to estimate hydraulic leakage and recommend maintenance and servicing tasks using aircraft standard sensors such as fluid temperature and reservoir level.

The proposed method was tested using several aircraft operating data with different levels of degradation (external leakages) and the results were analyzed in order to evaluate its precision on estimating leakage. Results showed the capability to detect leakage although uncertainties must be considered when evaluating maintenance interventions.

1. INTRODUCTION

Increased aircraft availability is one of the most desirable fleet characteristics to an airliner. Delays due to unanticipated system components failures cause prohibitive expenses, especially when these events occur on sites without proper maintenance staff and equipments. In recent years researches have focused on providing new technologies which could detect incipient failures and notify maintenance staff in advance when any component is about to fail. On the other hand these technologies requires several sensors that sometimes are not available on the aircraft which limits their application and consequently operational savings.

Hydraulic systems are found on most of the aircrafts nowadays and contain several components with significant failure rates. Some sensors are available to monitor them, but due to the number of components and their distributed localization along the aircraft, several faults are not monitored. Hydraulic fluid leakage is one example.

Wlamir Vianna et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Hydraulic leakage detection systems major applications are in the oil and gas industries (Stavenes, 2010) focusing most on pipelines such as "American Petroleum Institute Publ 1149" and (Beushausen, 2004). Aircraft applications are most of the times limited to visual inspections of some components with higher failure rates or some internal leakage monitoring such as pumps case drain flow monitoring as presented in (Copsey, 2006) and (Byington et al. 2003). The main issue related to aircraft applications is the sensors availability. Most of the aircraft hydraulic systems do not contain the proper set of sensors to monitor leakage although dispatch recommendations are made for leakage limits.

The method presented in this article describes a method to detect total system leakage using only a set of sensors available on most of aircrafts.

2. SYSTEM DESCRIPTION

A simplified architecture of aircraft hydraulic systems can be summarized as Figure 1.

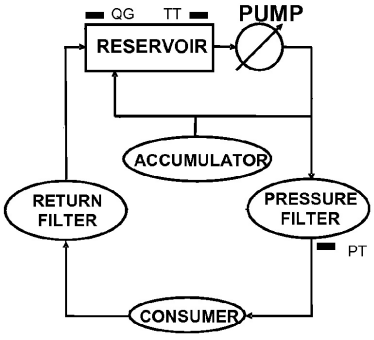


Figure 1 General Schematic of a Hydraulic System (Vianna, 2008).

The system contains one or more variable displacement pumps, accumulators, filters, and consumers, that include all the actuators connected to the hydraulic power such as flight controls, brake and landing gear. Also the system contains a bootstrap reservoir. The basic set of sensors available are pressure transducers (PT) at the pressure line, fluid temperature transducers (TT) at the reservoir and a quantity gauge (QG) indicating the reservoir level.

3. LEAKAGE DETECTION METHOD

The method here described was created for the EMBRAER Regional jets (E-Jets). On this platform the three sensors listed in Figure 1 were available and recorded on the Flight Data Recorder (FDR). Figure 2 illustrates some flight records for the reservoir level and fluid temperature under nominal behavior.

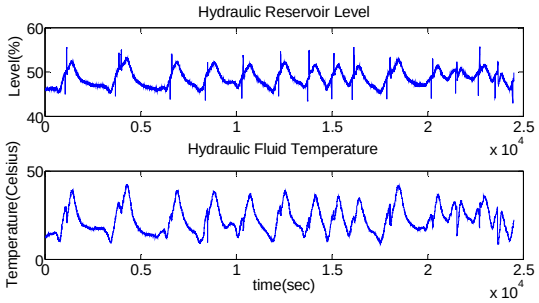


Figure 2 Hydraulic system flight record example.

From Figure 2 it is possible to conclude that direct measurement of the reservoir level is not enough to estimate the system total leakage since the fluid is submitted to a significant variance of temperature. Also some actuators (landing gear specially) interfere on the measured reservoir level as observed by the spikes in the first curve in Figure 1 when the landing gear is actuated.

The first step is to eliminate the influence of these parameters on the level measurement and to accomplish that a model was proposed considering fluid physical properties. According to (Merrit, 1967), a linear approximation for the fluid density is:

$$\rho = \rho_0[1 + \frac{1}{\beta}(P - P_0) - \alpha(T - T_0)] \quad (1)$$

where:

β is the Bulk Modulus

α is the Coefficient of Expansion

ρ_0 is the initial density (ISA Condition)

P_0 is the initial pressure of 1 atm (ISA Condition)

T_0 is the initial temperature of 15 ^{o}C (ISA Condition)

ρ is the actual density

P is the actual pressure

T is the actual temperature

For the elimination of temperature variation on the reservoir level, the volume was estimated for constant fluid density at ISA (International Standard Atmosphere) conditions. By manipulating Eq. (1), the hydraulic system fluid volume at ISA conditions is:

$$V_0 = V[1 + \frac{1}{\beta}(P - P_0) - \alpha(T - T_0)] \quad (2)$$

where:

V_0 is the hydraulic system fluid volume at ISA conditions

V is the hydraulic system fluid volume

The relation between V and the reservoir level indication is

$$V = V_{QG} + V_{sys} \quad (3)$$

where:

V_{QG} is the sensor indication

V_{sys} is the system volume excluding reservoir. It contains all volumes specified in "SAE Aerospace Standard AS5586"

To estimate V and consequently V_0, it is necessary to estimate V_{sys} first. Two methods could be used for that. The first one is to measure the volume of fluid necessary to fill the entire hydraulic system, and the second is to estimate V_{sys} by minimizing Eq. (4) using aircraft operating data (for example those in Figure 2) in a healthy condition. A gradient descent method was used to solve this equation.

$$ArgMin[\mathrm{var}(V_0), V_{sys}] \quad (4)$$

which is the same as:

$$ArgMin\left\{\mathrm{var}\left[V(1 + \frac{1}{\beta}(P - P_0) - \alpha(T - T_0))\right], V_{sys}\right\} \quad (5)$$

It was assumed V_{sys} constant, which in other words means that variances in actuators, piping, accumulators and any other components volumes were not considered . To minimize these variations only data with similar operating conditions (for example cruise) were used and with no observed leakage.

After estimating V_{sys}, the value of the estimated quantity gauge sensor indication at ISA condition (V_{est}) was estimated with Eq. (6) representing the mass estimation

(density multiplied by volume) for both temperatures: ISA and actual temperature.

$$\rho_0(V_{sys}+V_{est})=\rho(V_{sys}+V_{QG}) \quad (6)$$

For illustration purposes, the same data of Figure 2 was used to estimate the values of V_{est} (using Eq. 6) illustrated in Figure 3.

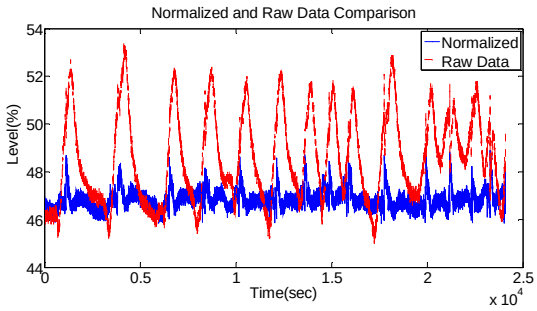


Figure 3 Normalized level indication (V_{est}) Vs Raw data level indication.

The variance of the raw data from Figure 3 was 3.41 and the variance of the normalized data was 0.157. Although reservoir level variance decreased significantly, some variations still persisted probably caused by non uniform fluid properties in the system and consumers' variations (accumulators for example).

If no data is available for fluid properties (β and α), a principal component analysis (PCA) could be used to eliminate the temperature influence (1st component). Figure 4 illustrate the relation between temperature and reservoir level for the same data in figure 2

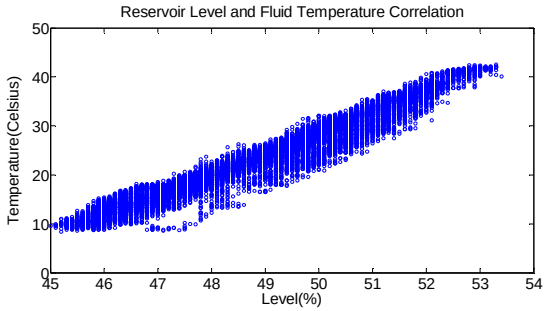


Figure 4 Relation between temperature and reservoir level.

The coefficients (loadings) of the two components are given by the following matrix:

$$\begin{bmatrix} 0.230 & 0.973 \\ 0.973 & -0.230 \end{bmatrix}$$

and the components variances:

$$\begin{bmatrix} 62.1 \\ 0.148 \end{bmatrix}$$

The much larger value of the first component variance indicates the strong correlation of level and temperature as expected.

The expected hydraulic system leakage can be determined through the angular coefficient of a linear interpolation of the normalized levels over the time. A least square method was used with data collected from the last 5 flights. Eq. (7) represents the equation variables estimated from the least square method.

$$Level(t)=(-Leakage)t+InitialLevel \quad (7)$$

4. SERVICING AND MAINTENANCE RECOMMENDATION

The current method triggers two possible maintenance actions. The first one is the inspection of the system and repair of leaking components when leakage estimation reaches a predetermined threshold. This task could be an improvement of the traditional periodical visual inspection. The next one is the reservoir hydraulic fluid filling service. This task can be trigged when for example the estimated future level for 5 days from now will reach the minimum allowed level to operate the system. This expected future level can be obtained from Eq. (7).
By using both of these alerts, maintenance could improve leakage inspections and optimize filling services, reducing non-schedule maintenance activities and AOG (Aircraft On Ground) events.

5. RESULTS

To validate the method operational data were used. Several flights from different aircrafts were collected and analyzed under several different health conditions. Figure 5 illustrates the three main different situations observed from all those data. Each sample represents the average level for 1 flight.

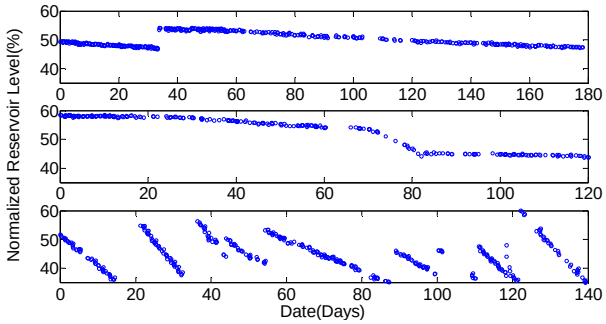


Figure 5 Examples of normalized level estimations.

The upper example shows an aircraft with no significant leakage as the reservoir level decreasing rate (leakage) is

low. Also it was possible to observe a filling task around day 35 (abrupt increase in level).

The middle example shows a failure around day 70 and its repair around day 81, probably detected from visual inspection.

The lower example shows a system with increased leakage requiring several hydraulic filling tasks in order to keep the system within the required levels. Probably the visual inspections executed for this example could not detect the excessive leakage.

For the same examples the leakage was plotted and displayed in Figure 6.

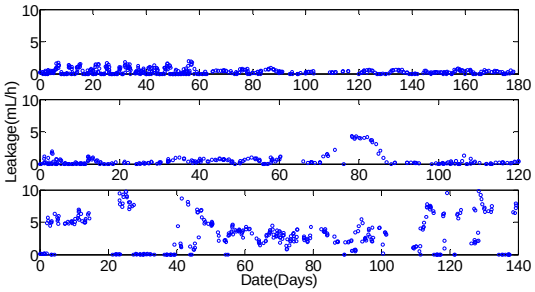


Figure 6 Examples of leakage estimations.

It is possible to observe that leakage estimations are noisier than levels estimations, especially with the presence of higher levels of leakage as seen in the third example of figure 6. This behavior is caused by the derivative nature of leakage estimation when few errors in level estimation generate increased errors in the leakage (derivate). One possible solution to minimize this error is to increase the interpolation window, here established in 5 flights. Although it softens the results, it increases the time response of leakage detection.

From all flights analyzed, 1202 filling tasks were executed in which 541 could be eliminated if the proposed method were used. Also a histogram is plotted in Figure 7 showing the leakage estimation for all flights analyzed.

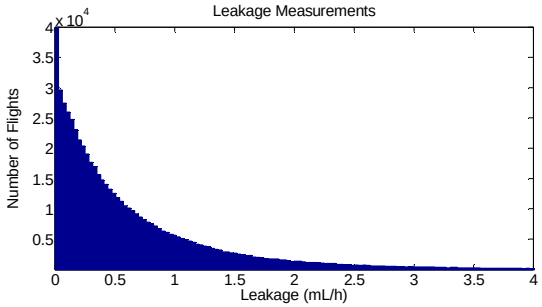


Figure 7 Fleet leakage estimation histogram.

From this plot, it is possible to perform several statistical analysis for the entire fleet and each individual aircraft such as an estimative of the number of flights with leakage levels above the recommended limit and how each aircraft is positioned compared to the entire fleet.

6. CONCLUSION

Aircraft hydraulic leakage detection maintenance tasks are time consuming and often do not bring an estimation of the leakage of the entire system. Also the lack of dedicated sensors makes this estimation more difficult. This paper presented a method to estimate total leakage and future reservoir levels from a hydraulic system considering only reservoir quantity gauge, fluid temperature and fluid pressure sensors. Also servicing and maintenance recommendations were proposed for these estimations in order to increase fleet leakage detection and reduce AOG (Aircraft On Ground) events.

Several aircraft data were used to validate the method. Although some estimations were less precise (leakage estimation), the method showed to be promising.

ACKNOWLEDGEMENT

The research reported here is supported by FINEP (Research and Projects Financing) under contract number 1498/07. FINEP is a publicly owned company subordinated to the Brazilian Ministry of Science and Technology (MCT), and is the Executive Secretary of FNDCT (Funding for Scientific and Technological Development).

REFERENCES

Stavenes, T., (2010), *Subsea Hydraulic Leakage Detection and Diagnosis* M.Sc. Thesis. Norwegian University of Science and Technology. Trondheim, Norway

American Petroleum Institute (API) (1993), Pipeline Variable Uncertainties and Their Effects on Leak Detectability. In API Publ 1149 1st edition Washington D.C.

SAE Aerospace Standard (2005), General Requirements for Hydraulic System Reservoirs. *AS5586* Issued 2005-02: SAE International.

Beushausen, R. (2004), Transient Leak Detection in Crude Oil Pipelines, *International Pipeline Conference*, October 4-8 Calgary, Alberta, Canada.

Merrit, H. E., (1967), *Hydraulic Control Systems*. New York: John Willey & Sons.

Vianna, W. O. L. (2008), *Modelagem e Análise do Sistema Hidráulico de uma Aeronave Comercial Regional*. M.Sc. Thesis. Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil.

Copsey R. (2006), Case Drain Monitoring: Part of a Healthy Recipe to reduce Downtime. *Hydraulics & Pnaumatics Magazine February* 2006

Byington, C. S.; Watson, M., Edwards, D. and Dunkins, B. (2003). In-line health monitoring system for hydraulic

pumps and motors. *IEEE Aerospace Conference Proceedings*, Big Sky, MO.

BIOGRAPHIES

Wlamir Olivares Loesch Vianna holds a bachelor's degree on Mechanical Engineering (2005) from Universidade de São Paulo (USP), Brazil, and Master Degree on Aeronautical Engineering (2007) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2007. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications in aeronautical systems.

João Pedro Pinheiro Malère holds a bachelor's degree in Control Engineering from Universidade Estadual de Campinas (Unicamp, 2004), Brazil, and a Master Degree in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, 2007), São José dos Campos, São Paulo, Brazil. He is with EMBRAER S.A., São José dos Campos, São Paulo, Brazil, since 2006. He works as a Development Engineer in an R&T group at EMBRAER performing research on IVHM technology for application to aeronautical systems. His current research interest is on integrated health management systems.

Dependable Wireless Sensor Networks for Prognostics and Health Management: A Survey

W. Elghazel[1], J.M. Bahi[2], C. Guyeux[2], M. Hakem[2], K. Medjaher[1], and N. Zerhouni[1]

[1] *Automatic Control and Micro-Mechatronic Systems Department, FEMTO-ST Institute*
Université de Franche-Comté, Besançon, France
wiem.elghazel@femto-st.fr
kamal.medjaher@ens2m.fr
noureddine.zerhouni@ens2m.fr

[2] *Computer Science and Complex Systems Department, FEMTO-ST Institute*
Université de Franche-Comté, Besançon, France
jacques.bahi@univ-fcomte.fr
christophe.guyeux@univ-fcomte.fr
mhakem@femto-st.fr

ABSTRACT

Maintenance is an important activity in industry as it reduces costs and enhances availability. This can be done either to revive a system/component or to prevent it from breaking down. The increasing need for reliability has led maintenance strategies to evolve from corrective to condition-based and predictive maintenance. The key process of the latter is prognostics and health management, a tool that predicts the remaining useful life of engineering assets. As plants are requested to offer both safety and reliability, planning a maintenance activity requires accurate information about the system/component health state. Usually, this information is gathered through independent sensors or a wired network of sensors. The use of a wireless sensor network has many advantages. First of all, the absence of wires gives sensor networks the ability to cover a large scale surveillance area. Second, it has become possible to monitor hostile and inaccessible areas by simply dropping the sensors from an aircraft to the monitoring region. Finally, the accuracy of measurements can be improved as the sensors can be placed at specific locations without being wired. Even though the deployment of wireless sensor networks is gaining great importance in monitoring applications, there are some research issues that still need to be studied to provide more accurate and reliable data. Indeed, we strongly believe that a good prognostic process starts with a reliable source of information; the wireless sensor network in our case. For this matter, in this paper, we discuss the dependability of wireless sensor networks, we highlight the attributes that have an impact on data accuracy, and present the state of the art in prognostics.

Wiem Elghazel et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Industrial systems are subject to failures, which can be irreversible or result in consequences varying from minor to severe. From this context, it is important to monitor a system, assess its health, and plan maintenance activities to avoid "catastrophic" failure results.

The research in Prognostic and Health Management (PHM) field has led to the development of prognostic models in an attempt to predict the Remaining Useful Life (RUL) of machinery before failure takes place. A maintenance schedule is then decided and system shutdown is prevented. Yet, if the prediction model and the provided measurements are not accurate, it is possible that the maintenance activity will be performed either too soon or too late.

Such a prediction activity requires online measurements of the operating conditions of the system under consideration. This information is usually gathered by the means of sensor nodes. In this study, we consider the case where the nodes communicate their information within a Wireless Sensor Network (WSN). Nevertheless, a WSN is prone to failure due to the nature of communication in the network and to the characteristics of its devices. For this reason, before deployment, a prior dependability study of the network is needed. It is the only way to guarantee the reception of accurate data.

Although both dependability of WSNs and prognostic models development have been studied and reported in the literature, as far as we know, none of the existing research work has

considered the dependability of WSNs for PHM purposes. In real life applications, the provided data can be inaccurate and incomplete. If this is not taken into consideration while building the prognostic model, the provided results cannot be reliable. Considering the limited computational capacities of WSNs, it is very common to privilege some dependability issues over others, regarding the target applications requirements. Thus, it is crucial to consider a "prognostic-oriented" dependability solution for WSNs.
This paper presents dependability issues with WSNs, that are relevant for RUL prediction, and discusses different prognostic approaches. The remainder of the paper is structured as follows. Section 2 presents an overview of wireless sensor networks. A state of the art in prognostics and health management is provided in Section 3. The relation between prognostics and WSN dependability and the remaining challenges are illustrated in Section 4. Finally, a conclusion is given in Section 5.

2. OVERVIEW OF WIRELESS SENSOR NETWORKS

WSNs are event-based systems that rely on the collective effort of several microsensor nodes (Akan & Akyildiz, 2005). This offers the network greater accuracy, larger coverage area, and the possibility to extract localized features. Typically, a WSN is composed of few base stations and hundreds (or thousands) of sensor nodes. A sensor node is a tiny device having the capability of sensing new events, computing the sensed values, and communicating information. Thus, the network can be deployed to monitor physical and environmental phenomena such as temperature, vibrations, light, humidity, etc.
There are different settings for a WSN model, which is generally dynamic, as radio range and network connectivity evolve over time. A network model can be either hierarchical, distributed, centralized, heterogeneous, or homogeneous (Z. Li & Gong, 2011).

2.1. Shortcomings of a WSN

WSNs are designed for an efficient event detection. They consist of a large number of sensor nodes deployed in a surveillance area to detect the occurrence of possible events. Such an activity necessitates efficiency, which is hard to achieve with the constraints of WSNs.
Available energy is a big limitation to WSN capabilities. In fact, sensor nodes are small sized devices, resulting in tiny and non-refillable batteries as energy supply (Carman, Kuus, & Matt, 2000). Moreover, wireless networks are vulnerable and necessitate security codes. Yet, processing security functions, transmitting security related data, and securing storage necessitate extra power, which is critical for WSNs (Carman et al., 2000; Walters, Liang, Shi, & Chaudhary, 2007).
The wireless communication between sensor nodes renders packet loss highly probable. The absence of physical connections in the network can result in channel errors, missing

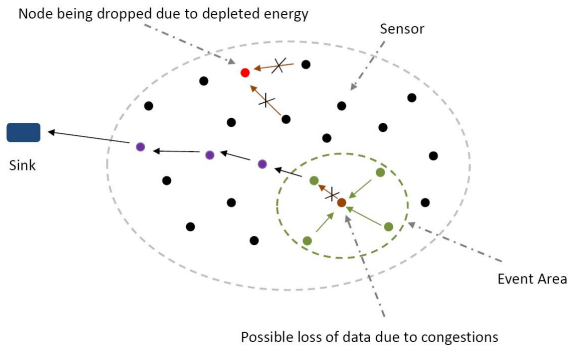


Figure 1. Illustration of some link failures in a WSN

links, and network congestion and cause packet drops. In addition to this, multi-hop routing and node processing lead to great latency and transmission errors in the network.
External deployment conditions also add to network vulnerability. WSNs are often deployed in harsh environments where they can be exposed to adversary attacks. Such attacks can cause permanent damage to the hardware. Thus, the network will remain unable to fulfill the intended tasks (Walters et al., 2007). Since the network is managed remotely, the sensor nodes are left unattended for a long period. It is yet impossible to detect physical tampering and to perform regular maintenance.
In Figure 1, two possible causes of packet loss are illustrated. In the first case, a previously established link between the sensors is lost. Once the parent node exhausts its energy, it is dropped from the network. As a result, a child node can no longer forward the sensed data and the previously received packets are permanently lost. In the second case, more than one sensor node simultaneously try to send data packets to the same parent, resulting in a network congestion and a possible loss of all the packets being forwarded at that level.
Considering all the limitations mentioned above, it is not easy for the network to always fulfill the intended tasks. Reliability and efficiency of WSNs are dependent on key issues, which are enumerated in the following.

2.2. Dependability issues

Sensor nodes have a short radio range and they collaborate to cover a given surveillance area. At the setup phase, it is crucial to ensure that the network covers the whole area (Tian & Georganas, 2005). The coverage problem arises as: "how to ensure that, at any time, any zone in the network is covered by at least one sensor node?"
Zorbas *et al.* (Zorbas, Glynos, & Douligeris, 2007) presented B{GOP}, a centralized coverage algorithm for WSNs. The algorithm proposes sensor candidate and avoids double-coverage depending on the coverage status of the corresponding field.
In (X. Wang et al., 2003), Wang *et al.* presented a protocol that can dynamically configure a network to achieve guaranteed degrees of coverage and connectivity. They gave a proof

that sensing coverage range does not need to be more than half the connectivity range in the network. Thus, their protocol helps preserve energy while maintaining coverage in the network.

As discussed before, available energy is a big limitation to WSNs. In order to prolong the network's lifetime, a possible solution is to keep a minimum number of sensor nodes in active mode. As WSNs rely on nodes density in the sensing and communicating processes, it is very likely that some nodes will not be needed. If a reliable node can forward data packets toward the sink, its neighbors can switch to idle state temporarily. Lifetime optimization using knowledge about the dynamics of stochastic events has been studied in (He, Chen, Li, Shen, & Sun, 2012). The authors presented the interactions between periodic scheduling and coordinated sleep for both synchronous and asynchronous dense static sensor network. They show that the event dynamics can be exploited for significant energy savings by putting the sensors on a periodic on/off schedule. The authors in (Kasbekar, Bejerano, & Sarkar, 2011) leverage prediction to prolong the network life time, by exploiting temporal-spatial correlations among the data sensed by different sensor nodes. Based on Gaussian Process, the authors formulate the issue as a minimum weight submodular set cover problem and propose a centralized and a distributed truncated greedy algorithms (TGA and DTGA). They prove that these algorithms obtain the same set cover.

As sensor nodes periodically go to sleep, they need to be awake when they are requested to. This is done by the transmission of wake-up messages towards a target sensor. However, if the message is not received at the right moment, data packets will be dropped. This will cost the network extra energy due to packet retransmission (Ye, Zhong, Cheng, Lu, & Zhang, 2003; Gallais, Carle, Simplot-Ryl, & Stojmenovic, 2006; J. Bahi, Haddad, Hakem, & Kheddouci, 2011).

In WSN, if the wear-out failures are not taken into consideration during the execution of the involved application, some nodes may age much faster than the others and become the reliability bottleneck for the network, thus significantly reducing the system's service life. In the literature, this problem has been formulated and studied in various ways. For instance, prior work (He, Chen, Li, et al., 2012; He, Chen, Yau, Shao, & Sun, 2012; Kasbekar et al., 2011) in lifetime reliability assumes node's failure rates to be independent of their usage times. While this assumption can be accepted for memoryless soft failures, it is obviously inaccurate for the wear-out-related fail-silent (a faulty node does not produce any output) and fail-stop (no node recovery) failures, because the sensor node's lifetime reliability will gradually decrease over time. To cope with this problem, a distributed self-stabilizing and wear-out-aware algorithm is presented in (J. M. Bahi, Haddad, Hakem, & Kheddouci, 2013). This algorithm seeks to build resiliency by maintaining a necessary set of working nodes and replacing failed ones when needed. The proposed protocol is able to increase the lifetime of wireless sensor networks, especially when the reliability of sensor nodes is expected to decrease due to use and wear-out effects.

2.3. Attacks in WSNs

WSNs suffer from limited computation capabilities, a small memory capacity, poor energy resources, absence of infrastructure, and susceptibility to physical capture. A variety of security solutions exists for infrastructureless networks (Ad hoc networks). Yet, they do not all answer the security challenges of WSNs.

WSNs are vulnerable to many attacks, due to their uncontrolled environment of deployment, the limitation of their resources, and the broadcast nature of transmission medium. The attacks are mainly classified under two categories: physical attacks and non-physical attacks.

Examples of well-known non-physical attacks in WSNs are: Denial of Service (DoS) attack, (Walters et al., 2007; Wood & Stankovic, 2002; Kim, Doh, & Chae, 2006), sybil attack, (Walters et al., 2007; Douceur, 2002; Zhang, Wang, Reeves, & Ning, 2005), traffic analysis attack, (Walters et al., 2007; Deng, Han, & Mishra, 2004), and node replication attack (Walters et al., 2007; Parno, Perrig, & Gligor, 2005; Braginsky & Estrin, 2002).

2.4. Dependability of WSNs

The dependability of a WSN is a property that integrates the attributes needed for the application to be justifiably trusted. Such a network should be able to deliver a correct service -a service that implements the system function- and makes sure that a failed component will not lead to system failure. System dependability was defined by Avizienis in (Avizienis, Lapire, & Randell, 2000) as "the ability of a system to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the users".

Developing a dependable WSN starts with defining the dependability requirements of users. In order to satisfy these needs, it is crucial to understand what might stop the network from delivering a correct service. In the following, we enumerate the attributes of a dependable network.

2.4.1. Availability

In the classical definition, a network is considered as highly available if its downtime is very limited. This can be due either to few failures, or to quick restarts when failures take place (Knight, 2004; Taherkordi, Taleghan, & Sharifi, 2006). If we add the security aspect, we can define availability as readiness for correct service for authorized users. This attribute can be computed as the probability that the network is functioning at a given time (Silva, Guedes, Portugal, & Vasques, 2012).

2.4.2. Reliability

A reliable network is a network that is able to continuously deliver a correct service. It can also be defined as the probability that a network functions properly and continuously in a time interval (Silva et al., 2012; Taherkordi et al., 2006).
Most of research works that have been accomplished so far employ retransmission mechanisms over redundancy schemes to achieve network reliability (Silva et al., 2012). The main purpose of a WSN is the correct delivery of data packets from sensor nodes to end user. Thus, reliability of WSNs is highly related to data transport. Reliability can be classified into different levels: packet reliability, event reliability, Hop-by-Hop reliability, and End-to-End reliability.
Both packet and event reliability levels deal with the required amount of information to notify the sink of the occurrence of an event within the network environment. Whereas the remaining two levels (i.e., Hop-by-Hop and End-to-End reliability levels) are concerned with the successful recovery of event information. Yet, all of them rely on retransmission and redundancy mechanisms.

2.4.3. Security

WSNs are different from traditional computer networks. Therefore, existing security mechanisms are not suitable for these networks. Developing adequate security measures requires understanding WSNs constraints related to security issues.
An attack on a network can be extended to more than just modifying the data packets originally circulating in the network. An attacker can inject additional data packets to disturb the normal function of the network and tamper with the decision making process. For this reason, a receiver (i.e., node) must be sure that the data being accepted is coming from a member of the network. Similarly, a sender needs to verify that the reception entity is whom it claims to be. This finality can be achieved through authentication.
Benenson *et al.* based their entity authentication on elliptic curve cryptography (Benenson, Gedicke, & Ravivo, 2005). Each user holds a legitimate certificate, which is the public key signed by a certification authority. Every node can verify the legitimacy of the users since the public key with the signature are preloaded in the sensors. Yet, this scheme requires an significant overhead for data encryption.
One of the most important issues related to network security is data confidentiality, and it refers to limiting data access to legitimate destinations. Keeping data packets confidential mainly means that:

- Sensor readings can only be performed by the legitimate destination; a sensor node holding information must not leak information to its neighbors.
- Communication channel has to be secured, especially when the data being communicated is highly sensitive.
- The network needs to achieve confidentiality by encrypting data during transmissions.

In (J. M. Bahi, Guyeux, & Makhoul, n.d.), Bahi *et al.* argue that in-network communication, node scheduling, and data aggregation need to be proven as secure. For this matter, they proposed a security framework for wireless sensor networks. The authors proved that in-network communication answers to security objectives (indistinguishability, non-malleability, detection resistance). In addition to this, the proposed algorithm is able to aggregate data over encrypted packets.

2.4.4. Defensive measures

Key establishment techniques have received great attention for many years. Nevertheless, WSN applications are relatively recent. Besides, the features of these networks are different from traditional networks. Therefore, preexisting techniques for key establishment are an unsuitable solution for WSNs applications. Traditionally, key exchange techniques use asymmetric cryptography (public key cryptography). Unfortunately, low power WSNs are unable to handle such a computationally intensive technique.
The easiest way for encryption keys distribution, is to establish one single key for the entire network and forward it. It is easy to notice that this method is inefficient as one node can compromise the entire network.
An alternative solution that can be adopted is symmetric encryption key. This technique secures communication between two hosts as they share a private key that is not recognized by the rest of the network. This key will be used for both data encryption and decryption.
Another possibility is random probabilistic key distribution scheme. The initialization stage starts with preloading in every sensor node a maximum number of keys (with respect to the memory). This is done in a way that two sets of keys (in two different nodes) will at least share one key. By broadcasting the identity of the keys, every node can discover the neighbors with which it can exchange information. Now, every node can only communicate with its legitimate neighbors; a link only exists between nodes sharing a key. It is now possible for a sensor node to safely establish a link with a target node by secretly sharing a key via their neighbors (Z. Li & Gong, 2011).

3. PROGNOSTICS AND HEALTH MANAGEMENT: STATE OF THE ART

Maintenance is an important activity in industry. It is performed either to revive a machine/component, or to prevent it from breaking down. Different strategies have evolved through time, bringing maintenance to its current state. This evolution was due to the increasing demand of reliability in industry. Nowadays, plants are required to avoid shutdowns while offering both safety and reliability (Peng, Dong, & Zuo, 2010).
The first form of maintenance is corrective maintenance. In

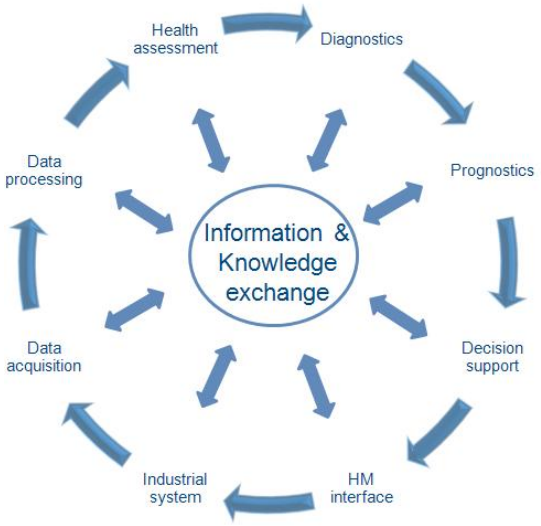


Figure 2. CBM Flowchart

this strategy, actions are only taken when the system breaks and can no longer perform the intended tasks. Yet, plants cannot afford to undergo breakdowns; in fact, sudden shutdowns cost money and time, in addition to safety and clients' trust. As a remedy to this problem, maintenance became a periodic activity. Domain experts rely on their knowledge and the observation of upcoming events to set time intervals in which the components are inspected and replaced if needed. This preventive (often called periodic) maintenance is especially adopted by transportations and nuclear plants (Hu, Youn, Wang, & Yoon, 2012). The main drawback of preventive maintenance is the fact that it is performed regardless of the machine's condition. In other words, industrials have to hire domain experts in order to set intervals for maintenance. Sometimes, this is unnecessary as the machine can be in a healthy state and this will cost extra and avoidable fees. Besides, even with periodic maintenance and inspections, random failures still occur. This is why Condition Based Maintenance (CBM) was proposed and developed in early nineties (Heng, Zhang, Tan, & Mathew, 2009).

CBM is a proactive precess for maintenance scheduling, based on real-time observations. It is an online model that assesses machine's health through condition measurements. As any maintenance strategy, CBM aims at increasing the system reliability and availability. The benefits of this particular strategy include avoiding unnecessary maintenance tasks and costs, as well as not interrupting normal machine operations (Heng et al., 2009).

In order to be efficient, a CBM program needs to go through the steps illustrated in Figure 2 (Jardine, Lin, & Banjevic, 2006).

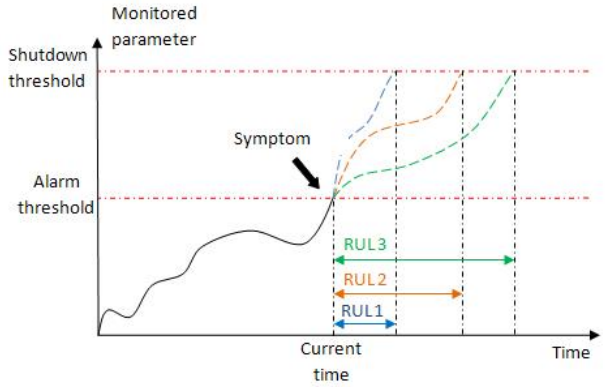


Figure 3. An illustration of RUL with uncertainties

3.1. PHM: definitions

The terms diagnostics and prognostics are widely used. Though, the difference between these two concepts is sometimes vague. However, it is important to specify the difference as it is the key to perform a good PHM.

PHM is the core activity of CBM, and it implies the same steps, namely: data processing, health assessment, diagnostics, prognostics, and decision making support.

While diagnostics aims at identifying and quantifying an actual failure, prognostics have the goal of anticipating failures. Several definitions concerning prognostics exist in the literature (ISO13381-1, 2004; D. Tobon-Mejia, Medjaher, & Zerhouni, 2012; D. A. Tobon-Mejia, Medjaher, Zerhouni, & Tripot, 2012; Zio & Maio, 2010; Jardine et al., 2006). Prognostics considers past events, the machine's current state, and operating conditions to estimate the Remaining Useful Life (RUL). This estimation is done by inspecting the evolution of continuous measurements of parameters that need to be monitored in time to assess the machine's state. These parameters can be temperature, humidity, vibration, pressure, and so on. A monitored parameter has a fixed threshold. Once reached, an alarm goes off indicating that a symptom of system deteriorating has been detected. The RUL is then computed with an associated confidence limit. The latter information illustrates to what point the predictions are trustworthy. The uncertainties of the RUL predictions have two causes: either the threshold value of monitored parameter, or the RUL prediction itself.

In Figure 3, we can observe the uncertainties that can be related to RUL prediction.

In (ISO13381-1, 2004), the necessary pre-requisites for reliable prognostics are proposed.

3.2. Classifying approaches

Prognostics approaches are classified under groups employing, more or less, the same techniques. Nevertheless, researchers use different classifications (Jardine et al., 2006),

(Heng et al., 2009), (Peng et al., 2010), (Sikorska, Hodkiewicz, & Ma, 2011), (Cadini, Zio, & Avram, 2009), (Hu et al., 2012), (D. Tobon-Mejia et al., 2012). More details on each approach can be found in the given references.
In this paper, we consider four groups: Physical models, Knowledge-based models, Data-driven models, and Hybrid models. They are detailed in the following sections.

3.2.1. Physical models:

Physical models rely on mathematical models to describe the physics of a failure, and are developed by domain experts. The first condition for a reliable model is a good understanding of the behavior of the system responding to stress. The description of the behavioral models is carried out via differential equations, state-space methods, or simulations.
Physical models are considered if:

- the mathematical model of the system is known;
- the failure mode is well understood;
- a physical model for each failure mode is available;
- the operating conditions can be monitored; and
- data describing the conditions related to each process isavailable.

Examples of model-based prognostics are given in (Y. Li, Billington, Kurfess, Danyluk, & Liang, 1999), (Byington, Watson, Edwards, & Stoelting, 2004), (Cempel, Natke, & Tabaszewski, 1997), (Qiu, Zhang, Seth, & Liang, 2002).

3.2.2. Knowledge-based models:

Since it is hard to build an accurate physical model for complex industrial systems, the employment of the latter is limited. Besides, it is impossible to apply a developed model to a different component. Other methods, such as knowledge-based ones, appear to be promising as they require no physical model.
In the following, two examples of this model are presented.

- Expert systems
  Since late 1960s, expert systems seemed to be suitable for problems usually solved by domain specialists. These models consist of computer system, designed to display expert knowledge. This knowledge is extracted by domain specialists and organized into rules learned by the computer to generate solutions.

  The rules have the form of:

  IF condition, **THEN** consequence

  Such a rule is strict and does not adapt to any changes in operating conditions. The only way to adapt the model to new situations is to add new rules whenever a new condition is observed. This can lead to a combinatorial explosion, given that a rule is required for every possible combination of inputs. Another limitation of this model is that it is only as good as its developers.

- Fuzzy logic
  It is a form of probabilistic knowledge, where the rules are approximate rather than fixed and exact. It was introduced by Zadeh in 1965 (Zadeh, 1965). The difference between fuzzy logic and classical predicate logic, is the use of fuzzy sets rather than discrete values standing for true or false. In a fuzzy set, variables membership is defined based on their degree of truth. The truth value ranges from 0 (completely wrong) to 1 (completely true).The rules may look like:

  IF condition “A” AND condition “B” **THEN** consequence.

  The description associated to the parameters differs from the description used with expert system rules. Here is an example to illustrate the difference:
  Expert system:

  IF engine is hot **THEN** shutdown

  Fuzzy logic:

  IF engine is slightly hot **AND** temperature is rising **THEN** cool down the system

  This new way of introducing rules gives the computer a very human-like and intuitive way of reasoning with incomplete, noisy, and inaccurate information. As a result, fault detection and prediction are more accurate, and for this reason, fuzzy logic is usually incorporated with other techniques.
  Even though this method can only be developed by domain experts, it is easy to understand the developed rules. It is not only recommended because it covers a large set of operating conditions, but also because of its efficiency when it is impossible to build a mathematical model or when data contains high levels of uncertainties and noise.

3.2.3. Data-driven models:

In data-driven approaches, models are directly derived from condition monitoring data, based on statistical and machine learning techniques. These models have a double role: assess current operating conditions and predict the RUL. Neither human expertise nor comprehensive system physics are needed for the prognostic model building process.
A data-driven prognostic model transforms raw data provided by the monitoring system into useful information, which combined with historical records, helps building a behavioral model and performing predictions. The data-driven approach is popular and widely-used because it offers a reasonable tradeoff

between complexity and precision. This approach remains the best solution when obtaining reliable sensor data is much easier than constructing mathematical behavioral models. Nevertheless, accuracy depends on many factors.

- The training set: normally, an efficient training requires a large set of inputs. It is not easy to decide whether the amount of inputs we dispose of is enough for training a reliable model or not.
- Operating conditions: manufacturing conditions change all the time, so do the environmental and operational conditions. All these changes may lead to uncertainties in the predictions as they refer to new situations that may not be recognizable by the model.
- Sensory signals: the amount of effective sensory data available when prediction is performed has an impact on accuracy.
- Degradation trend: RUL prediction relies on historical data and past events. As shown in Figure 3, the prediction is an extrapolation of what we observe up to the present moment. If the degradation trend is highly similar to a trend the model can recognize, prediction can be accurate (and inversely so).

Examples of the developed methods reported in the literature are:

- Aggregate reliability functions (Crevecoeur, 1993), (Duane, 1964), (Goode, Moore, & Roylance, 2000)
- Artificial neural networks ANN (Huang et al., 2007), (Herzog, Marwala, & Heyns, 2009), (W. Wang, Golnaraghi, & Ismail, 2004)
- Autoregressive moving average ARMA (Wu, Hu, & Zhang, 2007), (Yan, Koc, & Lee, 2004)
- Bayesian techniques (Cadini et al., 2009), (Kallen & van Noortwijk, 2005), (Weidl, Madsen, & Israelson, 2005)
- Hidden markov and hidden semi-markov models (Bunks, McCarthy, & Al-Ani, 2000), (Baruah & Chinnam, 2005), (Medjaher, Tobon-Mejia, & Zerhouni, 2012)
- Proportional hazards models (Z. Li, Zhou, Choubey, & Sievenpiper, 2007), (Liao, Zhao, & Guo, 2006), (Makis & Jiang, 2003)
- Trend extrapolation (Batko, 1984), (Kazmierczak, 1983), (C.Cempel, 1987)

3.2.4. Hybrid models:

Usually, prognostic activity does not consider one parameter. The monitored parameters are diversified, making it hard to study failure behavior using only one model.

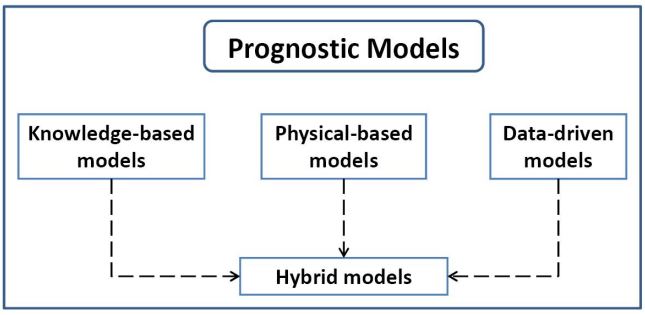


Figure 4. Categories for prognostic models

Hybrid models aim at improving prediction quality by providing more accurate RUL. All research works agree that physical models guarantee the most precise prediction. Nevertheless, even with good output quality, the complexity is too significant to ignore. This complexity can be reduced by adopting a data-driven approach. Thus, we can benefit from the merits of both prognostic approaches.
When physical understanding of failure mechanism and monitoring data are available, a hybrid approach is the best solution offering a compromise between model complexity and prediction accuracy.
In Figure 4 we illustrate the categories for prognostic models.

4. WSNS FOR INDUSTRIAL PHM AND CHALLENGES AHEAD

Reliability has become very essential in industry. It is a means to financial gain in addition to client trust. The research in the prognostic field, over the past years, resulted in a variety of tools and techniques offering plants the possibility to survey their systems, anticipate failures, and schedule maintenance. As the existent tools are different from one another, they have different advantages, drawbacks, complexities, etc. Data driven prognostic models drew a great deal of attention due to their low cost, low complexity, and easy deployment. The prediction model will first acquire information about the monitored system, assess the current state, and then extrapolate its future health state.
WSNs are mainly designed for surveillance purposes. They can be deployed in many fields such as military, automotive, agriculture, medicine...(Z. Li & Gong, 2011). Recently, industry has given WSN monitoring applications a great deal of attention. Nowadays, sensor networks are used to monitor industrial machinery for maintenance scheduling. The sensors deployed to survey the system/component will provide data to estimate the RUL. Nevertheless, if this data is inaccurate, the prediction based on it will not be relevant. The dependability requirements, discussed before, need to be considered before the network starts running. Thereby, they can provide accurate data for RUL prediction and maintenance scheduling. Despite the existence of many dependability solutions in WSNs, these solutions are not always applicable. As sensors

have restricted computational capabilities, solutions are often application oriented. Thus, a definition of dependability issues related to prognostics is essential.

Many aspects still need further studying in order to provide more reliable predictions. How can we explore available data? How can we consider operating conditions in RUL prediction? How can we allow multiple interactions while building a model? All these questions still need answers.

Data-driven models are designed to reduce model complexity and enhance real-time maintenance. For this matter, they only provide general predictions for a population of identical units; this makes prediction process easier and faster.

In the literature of prognostics, it is very common that the causes of a failure are limited to the values of monitored parameters. Other factors, although responsible for failures, seem to be neglected and overlooked. Although Condition Monitoring (CM) data reflects online monitoring, it does not replace reliability data. In fact, CM data provides measurements informing about a single component state at a specific moment. A failure does not only consider a single parameter (pressure, humidity...), it is a consequence of many factors (component age, different failing components...).

Reliability data, informing about all these factors, gives a bigger picture of the failing process. We are not neglecting the importance of CM data for prognostic process. However, while CM data provides information for short-term prediction, reliability data is able to extend these predictions until the next maintenance window. The complete neglection of operating conditions, operating age, and interactions between failures can only limit the application of developed models to real machines. Operating conditions are constantly changing, and if the model is unable to consider these changes, it is unable to produce reliable estimations. Furthermore, if we observe two similar components with different operating ages and operating under similar conditions, we will notice that they will not fail at the same time. Operating age definitely has an influence on time to failure. An internal failure can even accelerate or provoke another one.

Another issue to face while performing prognostics, is censored data. Many plants do not allow their system to run to failure. Components are often replaced before they actually fail. As a result, the real time to failure is not recorded. The performed preventive maintenance is mistaken for failure time, and RUL prediction is based upon that time. The value of RUL is critical for maintenance scheduling. In other words, the less accurate the prediction, the less reliable the maintenance schedule will be.

Maintenance scheduling is the reason behind building prognostic models. Once accomplished, the maintenance actions are not always considered in the model and generally, the related component is considered "as good as new". It is very important to consider the effects of maintenance actions in the prediction model, at least to evaluate the model efficiency and study the new failure behavior after the maintenance being performed.

What also drew our attention are the assumptions made to perform predictions. To the best of our knowledge, none of the previous research works has questioned the availability, safety, and security of data used for RUL prediction. It is assumed that:

- Sensory data is available and there is no data loss.
- The sensor network is reliable.
- There is no fault in the sensors.
- There is no constraint on energy consumption

So far, all prognostic work is limited to the condition monitoring layer, the health assessment layer, and the prognostic layer of the Open System Architecture for Condition-Based Maintenance OSA-CBM (Thurston, 2001; Niu & Yang, 2010). As RUL prediction concerns results that are yet to come, it has to rely on assumptions. Nevertheless, these assumptions, in no way, reflect a real life situation. The application of Wireless Sensor Networks (WSN) is very critical. First of all, the sensors size is very small. So they have very small batteries with limited disposable energy. If the communication in the network does not consider this limitation, the sensors will quickly consume all the energy they have and be dropped. Thus, the information will no longer circulate in the network. Still, an energy efficient WSN will not stop some nodes from being dropped. This means that the network has to be fault tolerant in order to be able to pursue its functionalities in case of any sudden events (sensor loss, interferences...). Besides, like all wireless networks, WSN can be hacked. Competitors and hackers can steal information, change data, cause damage to the system... Data circulating in the network needs to be secured against such attacks.

Many research works have been done in the field of WSN reliability. But every application has its own features, and generalized solutions do not always solve the problem. An adapted solution for prognostics needs and goals should be considered.

5. CONCLUSION

Condition-based maintenance is an important tool for modern plants in order to optimize their maintenance schedule. An appropriate schedule is reflected by the economical benefits. This paper went through the CBM process and its different steps leading to prognostics, and presented the different methods used in the literature of the latter to estimate the remaining useful life. Choosing one model over another mainly depends on (1) the available information to perform predictions and to study the systems behavior, (2) the complexity of the model, and (3) preferences regarding the domain of application, advantages and drawbacks of each model.

This paper also highlighted the fact that prognostic field still needs several improvements. RUL predictions cannot be ac-

curate if several points are not considered while building a model, namely (1) WSN dependability, (2) securing data, (3) including event data and censored data in the prediction process, and (4) model updates.

A discussion of dependability in WSNs is also given in this paper. In order to build a dependable network several attributes need to be considered: (1) network availability, (2) network reliability, and (3) network security. These attributes are the key for accurate data and reliable predictions.

REFERENCES

Akan, O. B., & Akyildiz, I. F. (2005, October). Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking*, *13*(5), 1003-1016.

Avizienis, A., Lapire, J.-C., & Randell, B. (2000). *Fundamental concepts of dependability* (Tech. Rep.). University of Newcastle.

Bahi, J., Haddad, M., Hakem, M., & Kheddouci, H. (2011). Distributed lifetime optimization in wireless sensor netwoks. In *Hpcc* (p. 432-439).

Bahi, J. M., Guyeux, C., & Makhoul, A. (n.d.). A security framework for wireless sensor networks: Theory and practice.

Bahi, J. M., Haddad, M., Hakem, M., & Kheddouci, H. (2013). Stabilization and lifetime optimization in distributed sensor networks. In *Aina* (pp. 437–442).

Baruah, P., & Chinnam, R. (2005, March). Hmms for diagnostics and prognostics in machining process. *International journal of Production Research*, *43*(6), 1275-1293.

Batko, W. (1984). *Prediction method in technical diagnostics*. Unpublished doctoral dissertation, Cracov Mining Academy.

Benenson, Z., Gedicke, N., & Ravivo, O. (2005). Realizing robust user authentification in sensor netwoks. In *Real-world wireless sensor networks (realwsn'05).*

Braginsky, D., & Estrin, D. (2002). Rumor routing algorithm for sensor netwoks. In *1st acm international workshop on wireless sensor networks and applications* (p. 22-31). NY, USA: ACM Press.

Bunks, C., McCarthy, D., & Al-Ani, T. (2000). Condition-based maintenance of machines using hidden markov models. *Mechanical Systems and Signal Processing*, *14*(4), 597-612.

Byington, C., Watson, M., Edwards, D., & Stoelting, P. (2004, March). A model-based approach to prognostics and health management for flight control actuators. In *Proceedings of the ieee aerospace conference* (Vol. 6, p. 3351-3362).

Cadini, F., Zio, E., & Avram, D. (2009). Model-based monte carlo state estimation for condition-based component replacement. *Reliability Engieneering and System Safety*, *94*, 752-758.

Carman, D. W., Kuus, P. S., & Matt, B. J. (2000, September). *Constraints and approaches for distributed sensor network security* (Tech. Rep.). NAI Labs, The Security Research Division, Network Associates, Inc. Glenwood.

C.Cempel. (1987). Simple condition forecasting techniques in vibroacoustical diagnostics. *Mechanical Systems and Signal Processing*, *1*, 75-82.

Cempel, C., Natke, H., & Tabaszewski, M. (1997). A passive diagnostic experiment with ergodic properties. *Mechanical Systems and Signal Processing*, *11*, 107-117.

Crevecoeur, G. (1993). A model for the integrity assessment of ageing repairable systems. *IEEE Transactions on Reliability*, *42*(1), 148-155.

Deng, J., Han, R., & Mishra, S. (2004). *Countermeasures against traffic analysis attacks in wireless sensor netwoks* (Tech. Rep.). University of Colorado.

Douceur, J. R. (2002, February). The sybil attack. In *Proceedings of the first international workshop on peer-to-peer systems (iptps'02).*

Duane, J. (1964). Learning curve approach to reliability monitoring. *IEEE Transactions on Aerospace*, *2*(2), 563-566.

Gallais, A., Carle, J., Simplot-Ryl, D., & Stojmenovic, I. (2006). Localized sensor area coverage with low communication overhead. In *Proceedings of the fourth annual ieee international conference on pervasive computing and communications* (pp. 328–337).

Goode, K., Moore, J., & Roylance, B. (2000). Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proceedings of the IMechE, PartE: Journal of Process Mechanical Engineering*, *214*(E2), 109-122.

He, S., Chen, J., Li, X., Shen, X. S., & Sun, Y. (2012). Leveraging prediction to improve the coverage of wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, *23*(4), 701-712.

He, S., Chen, J., Yau, D. K. Y., Shao, H., & Sun, Y. (2012). Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep. *IEEE Trans. Parallel Distrib. Syst.*, *23*(6), 1090-1102.

Heng, A., Zhang, S., Tan, A. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, *23*, 724-739.

Herzog, M., Marwala, T., & Heyns, P. (2009). Machine and component residual life estimation through the application of neural netwoks. *Reliability Engieneering and System Safety*, *94*(2), 479-489.

Hu, C., Youn, B. D., Wang, P., & Yoon, J. T. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engieneering and System Safety*, *103*, 120-135.

Huang, R., Xi, L., Li, X., Liu, C. R., Qiu, H., & Lee, J. (2007). Residual life prediction for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, *21*(1), 193-207.

ISO13381-1. (2004). *Condition monitoring and diagnostics of machines- prognostics- part1: General guidelines.*

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*, 1483-1510.

Kallen, M., & van Noortwijk, J. (2005). Optimal maintenance decisions under imperfect inspection. *Reliability Engieneering and System Safety*, *90*, 177-185.

Kasbekar, G. S., Bejerano, Y., & Sarkar, S. (2011). Lifetime and coverage guarantees through distributed coordinate-free sensor activation. *IEEE/ACM Trans. Netw.*, *19*(2), 470-483.

Kazmierczak, K. (1983). Application of autoregressive prognostic techniques in diagnostics. In *The vehicle diagnostics conference.* Tuczno, Poland.

Kim, M., Doh, I., & Chae, K. (2006, February 20-22). Denial of service (dos) detection through practical entropy estimation on hierarchical sensor networks. In *The 8th international conference advanced communication technology icact* (Vol. 3, p. 1566-1571).

Knight, J. C. (2004). An introduction to computing system dependability. In *Proceedings of the 26th international conference on software engineering (icse'04).*

Li, Y., Billington, S., Kurfess, T., Danyluk, S., & Liang, S. (1999). Adaptive prognostics for rolling element bearing condition. *Mechanical Systems and Signal Processing*, *13*, 103-113.

Li, Z., & Gong, G. (2011). *A survey on security in wireless sensor netwoks* (Vols. –; Tech. Rep.). Canada: Department of Electrical and Computer Engineering, University of Waterloo.

Li, Z., Zhou, S., Choubey, S., & Sievenpiper, C. (2007). Failure event prediction using the cox proportional hazard model driven by frequent failure signatures. *IIE Transactions*, *39*, 303-315.

Liao, H., Zhao, W., & Guo, H. (2006, January). Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In *Reliability and maintainability symposium* (p. 127-132).

Makis, V., & Jiang, X. (2003). Optimal replacement under partial observations. *Mathematics of Operations Research*, *28*(2), 382.

Medjaher, K., Tobon-Mejia, D., & Zerhouni, N. (2012, June). Remaining useful life estimation of critical components with application to bearings. *IEEE Transactions on Reliability*, *61*(2), 292-302.

Niu, G., & Yang, B.-S. (2010). Intelligent condition monitoring and prognostics system based on data-fusion strategy. *Expert Systems with Applications*, *37*, 8831-8840.

Parno, B., Perrig, A., & Gligor, V. (2005, May). Distributed detection of node replication attacks in sensor networks. In *Ieee sympsium on security and privacy.*

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *International journal of Advanced Manifacturing Technology*, *50*, 297-313.

Qiu, J., Zhang, C., Seth, B., & Liang, S. (2002). Damage mechanics approach for bearing lifetime prognostics. *Mechanical Systems and Signal Processing*, *16*, 817-829.

Sikorska, J., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*, 1803-1836.

Silva, I., Guedes, L. A., Portugal, P., & Vasques, F. (2012). Reliability and availability evaluation of wireless sensor netwoks for industrial applications. *Sensors*, *12*, 806-838.

Taherkordi, A., Taleghan, M. A., & Sharifi, M. (2006, December). Dependability considerations in wireless sensor netwoks applications. *Journal of Networks*, *1*(6), 28-35.

Thurston, M. (2001). An open standard for web-based condition-based maintenance systems. In I. S. R. T. Conference (Ed.), *Autotestcon proceedings* (p. 401-415).

Tian, D., & Georganas, N. D. (2005). Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Networks*, *3*, 744-761.

Tobon-Mejia, D., Medjaher, K., & Zerhouni, N. (2012). Cnc machine tool's wear diagnostic and prognostic by using dynamic bayesian networks. *Mechanical Systems and Signal Processing*, *28*, 167-182.

Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., & Tripot, G. (2012, June). A data-driven failure prognostics method based on mixture of gaussians hidden markov models. *IEEE Transactions on Reliability*, *61*(2), 491-503.

Walters, J. P., Liang, Z., Shi, W., & Chaudhary, V. (2007). Wireless sensor network security: A survey. In *Security in distributed, grid and pervasive computing* (p. 799-849). CRC Press.

Wang, W., Golnaraghi, M., & Ismail, F. (2004). Prognostics of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, *18*, 813-831.

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., & Gill, C. (2003). Integrated coverage and connectivity configuration in wireless sensor netwoks. In *First acm conference on embedded networked systems.*

Weidl, G., Madsen, A., & Israelson, S. (2005). Applications of object-oriented bayesian networks for condi-

tion monitoring, root cause analysis and desicion support on operation of complex continuous process. *Computers and Chemical Engineering*, *29*, 1996-2009.

Wood, A. D., & Stankovic, J. A. (2002, October). Denial of service in sensor networks. *Computer*, *35*(10), 54-62.

Wu, W., Hu, J., & Zhang, J. (2007). Prognostics of machine healt condition using an improved arma-based prediction method. In *Ieee* (p. 1062-1067). China.

Yan, J., Koc, M., & Lee, J. (2004). A prognostic algorithm for machine performance assessment and its application. *Production Planning and Control*, *76*, 796-801.

Ye, F., Zhong, G., Cheng, J., Lu, S., & Zhang, L. (2003). Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd international conference on distributed computing systems* (pp. 28–37).

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*, 338-353.

Zhang, Q., Wang, P., Reeves, D. S., & Ning, P. (2005). Defending against sybil attacks in sensor networks. In *25th ieee international conference on distributed computing systems workshops* (p. 185-191).

Zio, E., & Maio, F. D. (2010). A data-driven fuzzy appoach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engieneering and System Safety*, *95*, 49-57.

Zorbas, D., Glynos, D., & Douligeris, C. (2007). BGOP: An adaptive algorithm for coverage problems in wireless sensor netwoks. In *the 13th european wireless conference.*

Poster Papers

Cost-Benefit Analysis and Specification of Component-level PHM Systems in Aircrafts

Alexander Kählert[1], Sebastian Giljohann[2], and Uwe Klingauf[3]

[1,3] *Institute of Flight Systems and Automatic Control, Technische Universität Darmstadt, Darmstadt, Hessen, 64287, Germany*
kaehlert@fsr.tu-darmstadt.de
klingauf@fsr.tu-darmstadt.de

[2] *Lufthansa Technik, Frankfurt/M, Hessen, 60546, Germany*
sebastian.giljohann@lht.dlh.de

ABSTRACT

Unplanned aircraft groundtimes caused by component failures create costs for the operator through delays and reduced aircraft availability. Unscheduled maintenance on the other hand also creates costs for Maintenance, Repair and Overhaul (MRO) companies. The use of PHM is considered to improve the planning of component-specific maintenance and thus reduces consequential costs of unscheduled events on both sides.

This study assesses the component-specific costs and characteristics of today's maintenance approach. A discrete event simulation represents all relevant aircraft maintenance processes and dependencies. For this purpose the Event-driven Process Chain (EPC) method and Matlab/SimEvents are used. The data input (process information, empirical data) is provided by a particular MRO company.

Whereas recent approaches deal with stochastically processed data only, e.g. failure probabilities, the proposed method mainly uses deterministic data. Empirical data, representing particular dependencies, describes all relevant stages in the component lifecycle. This includes operation, line and component maintenance, troubleshooting, planning and logistics.

By simulating different scenarios, various maintenance future states can be evaluated by analysing effects on costs. The obtained economical and technical constraints allow to specify component-level PHM design parameters, as minimum prognostic horizon or accuracy. Detailed process-specific information is provided as well, e.g. costs of non-productive MRO activities or no-fault-found (NFF) characteristics.

Alexander Kählert et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Since the development in research fields as e.g. fuel efficiency has reached a point, where the savings potential is expected to advance incrementally only, the concept of PHM offers new opportunities to improve competitiveness, see (Sun, Shengkui Zeng, Kang, & Pecht, 2010) and (Feldman, Jazouli, & Sandborn, 2009). By converting unplanned aircraft groundtimes into planned maintenance tasks, it is considered to support the general objectives of aircraft maintenance. According to (Fromm, 2009) and (Knotts, 1999) these are:

- maximising aircraft availability and dispatch reliability
- minimising consequential costs of technical delays
- minimising direct maintenance costs (DMC)

The dispatch reliability (DR) describes the ratio of revenue departures without delays or cancellations compared to all flights. A higher DR results in a higher aircraft availability and thereby implies a reduction of delay compensation costs (e.g. rescheduling costs, payoffs) as well as lower opportunity costs through more revenue flights, see (Rodrigues, Balestrassi, Paiva, Garcia-Diaz, & Pontes, 2012) and (Sisk, 1993). According to (Eurocontrol, 2010) average costs of aircraft delays reach $113 per operating minute. Other results are given in (Rodrigues et al., 2012), (Cook, Tanner, & Anderson, 2004) and (Fritzsche & Lasch, 2012). In 2008 European airlines experienced 85 million delay minutes, creating estimated overall costs of $9.7 billion, see (Eurocontrol, 2011). According to (Eurocontrol, 2012) technically induced delays account for a large portion of all delays. Since PHM allows to perform maintenance tasks within planned maintenance events prior to a component failure, technical delay costs are one measured variable in this study. The scheduling is based on remaining useful life (RUL) prediction, e.g. see (Hölzel, Schilling, Neuheuser, Gollnick, & Lufthansa Technik AG, 2012). Within a prognostic horizon (PH) the future

system state can be predicted with sufficient accuracy.

Reducing the DMC, being part of the direct operating costs (DOC), is another key factor in competition. According to (Fromm, 2009) DMC account for approximately 7% of the DOC of an airline. (Fritzsche & Lasch, 2012) state that PHM enables a MRO company to optimise the maintenance program's scheduling as well as structuring. This allows more efficient processes by converting unplanned into planned activities and preventing non-productive events. Therefore avoidable maintenance costs are another measured variable in this study.

Since a PHM system is not ideal, it is characterised by uncertainties and involves various risks:

- The PH is too short and allows no useful forecast.
- A low accuracy causes misinterpretation (NFF or undetected events).
- The PHM system itself fails (e.g. sensor failure).

In order to facilitate the planning of maintenance events, the PH has to allow forecasts for a certain number of flight cycles (FC) or flight hours (FH). For instance, if a component malfunction is indicated by a RUL prediction 5 minutes prior to failure, it may not be early enough in order to prevent a groundtime at the next station. On the other hand 5 minutes might be enough to significantly improve operation in some cases (Sun et al., 2010). If the PHM system's accuracy is not sufficient, false conclusions are possible. Non-productive NFF events may be generated by false alarms, or unscheduled events by undetected failures, see (Knotts, 1999) and (Hölzel et al., 2012). Furthermore, a PHM system involves requirements concerning its own maintainability.

In summary, the goals of this study can be defined as follows:

1. Evaluate the financial potential of a component-specific PHM system.
2. Specify component-based PHM parameters.

The required component-specific PHM performance parameters, as PH and accuracy, can be specified by the evaluated economic constraints. These are gained from the calculation of a component-specific PHM system's effect on

- delay compensation costs and
- direct maintenance costs

with respect to the different PHM design parameters. Studies analysing similar goals often use simulation models. (Hölzel et al., 2012) employ a model to carry out a cost-benefit analysis by using failure probabilities as input data and evaluating savings potentials of different PHM systems. An alternative to the data input approach will be discussed in sections 2.2.1 and 2.2.2. Another similar procedure is presented in (Feldman et al., 2009). Key of this study is the Return on Investment (ROI) calculation. Component failures are generated probabilistically as well in this case.

Compared to the other studies, this paper presents a PHM evaluation using mostly deterministic data to simulate maintenance events as close to reality as possible. This is enabled by available, adequate MRO data. The methodology, including assumptions and limitations, is discussed in the next chapter.

2. METHODOLOGY

This chapter provides an overview of the applied approach, illustrated in Figure 1. The major steps described in the next sections are indicated by the labeled arrows: Data preprocessing, modelling and simulation. The boxes represent the in- and outputs, further explained in the particular sections.

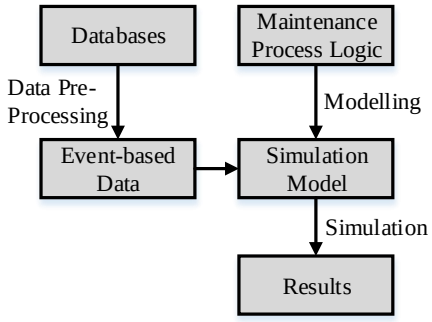


Figure 1. Description of the approach.

2.1. Component-Level Approach

The component-level approach used in this study is explained in the following.

2.1.1. Level of Detail

The introduced approach aims to evaluate the effects of a PHM system on component or line-replacable unit (LRU) level. LRUs are designed to be replaced quickly during turn around times between two flights. Hence, faulty LRUs are responsible for technical delays in many cases, because the replacement requires a prior diagnosis as well as spare parts and often takes place during flight operation time.

An evaluation on a more detailed level is not conducted, due to the fact that most available MRO data provides LRU specific data only. In most cases LRUs can be found within an ATA-6-digit chapter, specified as a subsystem. For further information on the ATA-numbering system see (Air Transport Association of America, 2012). If an LRU is supplied by different manufacturers or modified by minor updates, different partnumbers (PN) are applied.

2.1.2. Sample Component Selection

The assumptions made in this study require the evaluated LRUs to fulfill the following requirements:

- Standard LRU maintenance applies.
- LRU shows any sort of wear behaviour.
- LRU causes high costs through delays and cancellations.
- LRU causes high costs through NFF events.

It is assumed that all LRUs pass through a standardized LRU maintenance process, which is the focus of this study. The wear behaviour provides information about the technical feasibility of a prognosis application. In order to be able to perform prognosis an observable degradation process is necessary, whereas diagnosis requires the binary states "functional" and "not functional" only. LRUs can cause operational delays through time-consuming replacements or troubleshooting (TS) tasks. Costs through NFF events can result from insufficient fault interpretation and the conflicting goals of different maintenance departments. Line maintenance at an airport aims to assure an aircraft's availability by performing all tasks as quickly as possible, e.g. by simply replacing an LRU in case of a fault indication, even if a detailed TS was not conducted. The shop maintenance on the other hand overhauls all incoming LRUs. If a line maintenance replacement takes place without any exact finding, the subsequent shop maintenance event might be rated as NFF. This can be considered a non-productive maintenance action.

Besides the cost factors, the minimum equipment list (MEL) is considered for the LRU selection as well. A MEL category is specified by the corresponding rectification interval (RI) of a component or its function. The RI shows how urgently a problem has to be fixed in order to keep an aircraft realeased to service. Thus, a failure's priority and operational risk can be described. Examples for MEL RI are given in Table 1.

Table 1. MEL rectification intervals.

MEL RI	Time for rectification
A	instantly or failure-specific
B	within 3 days
C	within 10 days
D	within 30 days

In order to select adequate LRUs for the study, prior to the simulation all LRUs are ranked. Based on the available MRO data, a ranking as exemplarily shown in Table 2 for the LRU Air Data Inertial Reference Unit (ADIRU) is obtained. MRO component data from the years 2010 to 2013 is considered, providing estimated annual costs for delays and NFF events as well as the corresponding MEL RI categories for each LRU. At the end of this study the exemplary results for the ADIRU are discussed.

Table 2. Ranking of LRUs.

ATA	LRU	Delay costs	NFF costs	MEL RI
34-12-34	ADIRU 1	C_{Delay}	C_{NFF}	A
...	...	...	...	...

2.1.3. Component Maintenance

The LRU maintenance process can be described by the main modules shown in Figure 2. The interface between airline operation and the MRO involves the TS, the maintenance planning and the system maintenance. In the following the term system describes the aircraft, consisting of subsystems, the LRUs. The TS mainly derives supporting and maintenance actions from fault isolation, e.g. by interpretation of fault messages. The planning department concentrates on the time scheduling of maintenance tasks considering priority, required time as well as available ground times. The system maintenance consists of line and base maintenance. The subsystem (shop) maintenance, connected by the logistics, deals with the overhaul of LRUs. Repaired components are sent to and taken from the spare parts inventory. Since the impact of this study on the spares inventory is insignificant, it will not be subject in detail. Furthermore, information and material flows are illustrated.

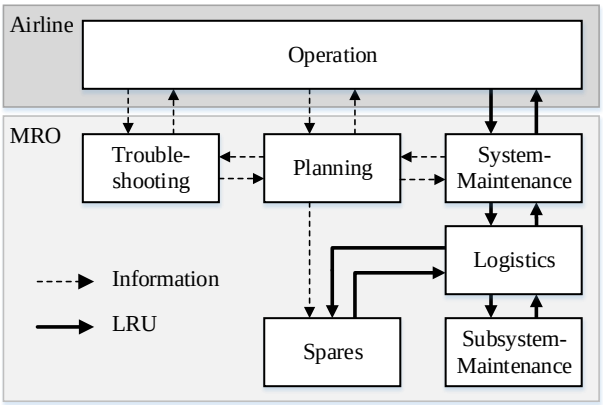


Figure 2. Modules of the component maintenance.

2.1.4. Non-routine Maintenance

The component maintenance can be subdivided into the fields routine and non-routine maintenance. Routine tasks deal with maintenance actions that are planned in advance. This applies for especially safety relevant items or consumables. Non-routine maintenance deals with unscheduled tasks, created by faults of components that are maintained on-condition. Since the earlier mentioned approach includes on-condition maintained LRUs only, this study focusses on non-routine maintenance. Furthermore, only maintenance events carried out at the homebase are analysed.

2.2. Discrete Event Approach

In a discrete event simulation (DES) state changes are only modelled at discrete time steps, called events. By skipping simulation times without any changes, the approach is very computing time-efficient. States are defined by objects, referred to as entities, and their attributes. Events are caused by attribute changes and the induced state transformations.

If a DES model uses non-probabilistic data only, it is called deterministic. Thus, all input variables are exactly defined and all states pre-determined. The use of a simulation model then primarily enables the computing of numerous operation steps. If input data is probabilistically specified, a simulation model allows to consider stochastic input by conducting a Monte Carlo simulation. A set of simulation runs then enables the representation of distributed variables.

DES allows to analyse interdependencies between particular events in detail, as described in (Rodrigues et al., 2012). For instance, information about failure message generation, LRU replacements and aircraft delays can exactly be represented and correlations described. Whereas pure probabilistic approaches mainly allow analysis concerning particular factors (consequence-wise analysis), an event-wise analysis provides information about specific causes and effects (see Figure 3). In this study both data input types, probabilistic distributions and deterministic data, are used.

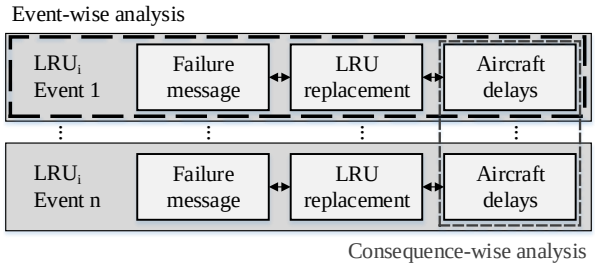


Figure 3. Different analysis approaches.

2.2.1. Stochastic and Deterministic Data

If particular data is not described by a constant value, it is distributed. According to (Kohn, 2005) probability density functions (PDF) allow to describe the probability of a value to apply. An example for uncertain data used in this study is varying process time. Since in reality not every LRU replacement needs the same amount of time, an analysis of past process durations provides statistical information on the empirical distribution. Figure 4 shows different PDF types. Depending on how accurate the empirical data is available, one of the introduced approaches is used. If only one scalar value is available, the special case deterministic distribution applies. This is the case for most input data in this study.

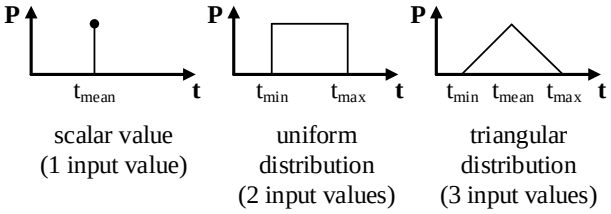


Figure 4. Used probability density functions.

2.2.2. Component Failure Generation

As opposed to many other studies, as (Hölzel et al., 2012) or (Feldman et al., 2009), the chosen approach defines component failures deterministically. Since empirical data regarding date and time of a component failure or replacement is available, all temporal information is inherited. Thereby different analysis scenarios all refer to the same initial failures as the root cause for replacements and allow exact comparisons.

2.2.3. Process Definition

In order to acquire knowledge about the overall maintenance process, a conducted process analysis provides information about the following process factors:

- work type (information-/LRU-processing)
- process time (minimum/average/maximum)
- number of required personnel
- qualification of required personnel
- required resources (e.g. hangar)

By mapping the process sequence including conjunctions, the process interdependencies are represented (see Section 2.4). Whereas the information on process sequence and personnel requirements is derived from MRO documents, the process times of LRU-specific processes are specified by maintenance experts and historical data. Concerning process resources only the demands are modelled as opposed to available capacities.

2.3. Data Acquisition and Preprocessing

The data preprocessing provides the event-based data input for the simulation. It is described in the following sections.

2.3.1. Input Data

Input data for the simulation is derived from various MRO databases. Flight log databases provide information about the flight schedule, ground events and operational irregularities. Fleet databases contain registration-specific information. A variety of technical logbooks provide data about failure messages, the maintenance history (reports and actions) and logistics. Experts contributed process-specific details.

All databases contain data sets that are exactly defined by the attributes aircraft registration, LRU part- and serialnumber,

date, time and location. According to the logic introduced in the next section, corresponding data sets from different databases are connected to single events.

2.3.2. Event Definition

An LRU replacement event is specified by data from the aforementioned databases. In order to identify and extract data event-wise, the linking logic, shown in Figure 5, is applied. (Beynon-Davies, 2004) further discusses data models.

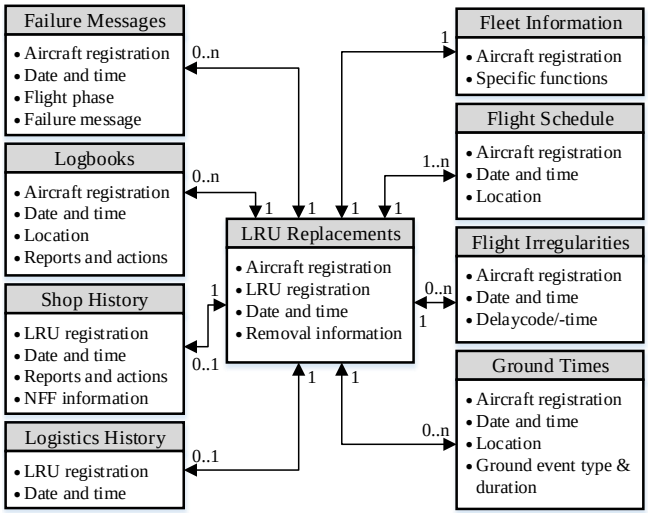


Figure 5. Relational object data model for an event definition.

As shown in the data model, an LRU replacement data set entry is the basis for an event definition. Based on the available attributes, all other databases are connected by linking parameters, e.g. aircraft registration and date. As indicated by the data model, several conjunction types are used. The connection of multiple data sets is possible (n) as well as single data entries or no data at all (1 or 0). By matching all relevant data, unique subsets specifying separate events are defined. Matching conflicts, redundancies or incomplete data is accounted for by robust merging, either correcting or skipping the particular data set. Insufficient data quality is a major limitation in this study. Therefore only reliably defined replacement events are considered for the evaluation.

The data is organised in the structure shown in Figure 6. Different hierarchy levels are used in order to classify similar information. Thereby results can later be analysed concerning particular characteristics, e.g. comparing all events of k different partnumbers for one LRU.

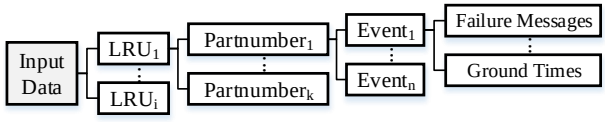


Figure 6. Hierarchy levels of the obtained data structure.

2.4. Modelling

The following sections explain the model building.

2.4.1. Process Modelling

The EPC method is used for the logical maintenance process modelling. It comprises the elements process, event and Boolean operators (AND, OR, XOR). A process, illustrated by a rectangle, is defined by the aforementioned process factors. An event, displayed as a hexagon, defines the state that is supposed to be reached after a process completion. The logical operators, illustrated by circles, enable the modelling of intersections by defining routing conditions. Information flow is indicated by dashed lines. Figure 7 shows an example:

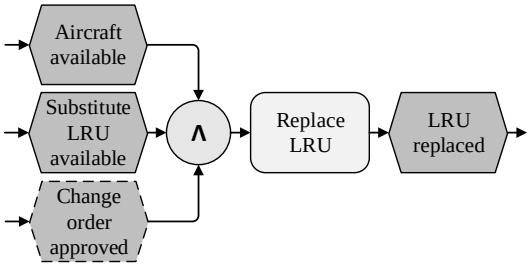


Figure 7. Example of EPC modelling.

By using the EPC method all modules of the component maintenance, shown in Figure 2, are described in detail. Due to intellectual property (IP) reasons, a detailed process map is not presented in this paper.

2.4.2. Simulation Model

The EPC model is transferred to a software model using Matlab SimEvents, as applied in (Gray, 2007) or (Bender, Pincombe, & Sherman, 2009). Matlab Stateflow is used to represent the system (aircraft) and subsystem (LRU) states. All defined states are shown in Table 3 and Table 4.

Table 3. System states.

z_{System}	**State description**
1	flight
0	on ground, other station
-1	maintenance, other station
-2	on ground, homebase
-3	unscheduled maintenance, homebase
-4	available for maintenance, homebase
-5	scheduled maintenance, homebase

An aircraft can only hold one particular system state at a time. Flight operation is represented by alternating system states $z_{System} \in \{-2, 0, 1\}$. Maintenance times are distinguished between scheduled $z_{System} \in \{-5, -4\}$ and unscheduled events $z_{System} \in \{-3, -1\}$.

Table 4. Subsystem states.

State	State description
1	regular operation
0	rectification in progress
-1	maintenance required
-2	deferred
-3	deferrable

An LRU holds the states functioning $z_{Subsystem} = 1$, in repair $z_{Subsystem} = 0$ or not properly functioning $z_{Subsystem} \in \{-3, -2, -1\}$, further described by the urgency accounted for by the MEL logic. Items that do not require immediate rectification, can be deferred. By defining discrete states and parameter dependent transitions, the toolbox allows to account for and evaluate different operating modes.

2.5. Simulation

The simulation characteristics are explained in the following sections.

2.5.1. Simulation Time Characteristics

One simulation cycle represents all events within the analysed time period for one aircraft at a time. This allows the evaluation of subsequent, interrelated events generated by different LRUs on the same aircraft. Due to computing time issues and the study objectives, only time frames of two weeks around an LRU replacement event are examined. Taking advantage of DES all dates without any relevant occurrences are skipped.

2.5.2. Scenario-based Analysis

If the degree of particular process transformations through PHM is supposed to be analysed, the definition of different simulation scenarios is useful. Defined scenarios are:

1. current state maintenance (data-based only)
2. best-case current state maintenance (data- / logic-based)
3. target state maintenance with PHM (data- / logic-based)

If the maintenance in its current state is to be analysed, the first scenario applies. In this case the simulation model directly uses the data input in order to represent all actions and queue times as they occurred in reality. The second scenario aims at the representation of a best-case evaluation of today's maintenance. The input data is used partially, e.g. date and time of first failure occurrence. The missing information is then generated by the modelled process logic. The third scenario is targeted on the evaluation of possible future states with PHM, by assessing the impacts of different prognosis parameters, as PH and accuracy. In this case only a small amount of the historical input data is used, e.g. first occurence of a failure message, in order to dissolve dependencies on today's procedure and to generate an ideal state case. The further rectification process is represented by the implemented process logic. By comparing the significant changes to possible maintenance characteristics with PHM, today's maintenance deficits can be analysed.

2.5.3. Monte Carlo Simulation

In order to account for input data provided as distribution functions, a Monte Carlo simulation carries out various simulation runs. Based on the in section 2.2.1 described distributions, at each cycle the stochastically provided input data is randomly assigned, creating slightly differing simulation results. This way especially the varying process times are accounted for. By defining and saving seed values - initial values for random number generators - all Monte Carlo simulation runs can be reproduced. The effects of the Monte Carlo simulation are considered in the model output interpretation by including the result's distributions and illustrating particular risks.

2.6. Target Values

The simulation results can be classified as process data and operational aircraft data. The results interpretation covers the statistical analysis of costs as well as raw, time-based simulation data. Cost values are obtained from calculation of time-based simulation data with available MRO cost rates. The simulations outputs are available on different levels of detail, allowing versatile result interpretation (see Figure 6). The different categories of target values are explained in the following sections. (Linser, 2005) e.g. gives an overview of other prevalent target values.

2.6.1. Costs

Cost analysis can be performed on all levels of detail. If desired, the IATA MRO cost structure, presented in (Fromm, 2009) or (Linser, 2005), can be considered. Primarily the approach determines costs for an event k according to the logic shown in eq. 1-3.

Event-based costs consist of process and operation irregularity expenses. Process costs are defined by labour, material and overhead expenses. Operational charges arise from flat rates defining compensation and opportunity costs of delays or Aircraft-on-Ground (AOG) times multiplied by the corresponding event duration.

$$C_{Event_k} = \sum_{i=1}^{m} C_{Proc_i} + \sum_{j=1}^{n} C_{Ops_j} \tag{1}$$

$$C_{Proc_i} = t_{L_i} \cdot n_{L_i} \cdot c_{L_i} + n_{M_i} \cdot c_{M_i} + C_{O_i} \tag{2}$$

$$C_{Ops_j} = t_{O_j} \cdot c_{O_j} \tag{3}$$

C_{Event_k}	Total cost of event k
C_{Proc_i}	Cost of process i
C_{Ops_j}	Cost of operational irregularity j
t_{L_i}	Process time
n_{L_i}	Amount of labour
c_{L_i}	Labour cost rate
n_{M_i}	Amount of material
c_{M_i}	Material cost rate
C_{O_i}	Overhead costs
t_{O_j}	Irregularity duration
c_{O_j}	Compensation cost rate

Future model updates will include ROI calculation, as described in (Feldman et al., 2009). This will enable the comparison of different scenarios concerning PHM investments and avoided costs.

2.6.2. Process Characteristics

The simulation output directly provides process-specific information, as time distributions and process sequences. By evaluating the raw data, non-monetary target values can be analysed. Some examples are:

- response and wait times
- time savings through process transformations
- process loops
- bottlenecks

2.6.3. Additional Results

Examples for parameters, relevant for the MRO company and not expressed as costs or process times, are:

- aircraft dispatch reliability and availability
- delay characteristics
- NFF characteristics
- effectiveness of actions
- real-time data transmission benefit

Regarding a PHM design the following prognosis parameters are evaluated:

- minimum required PH
- minimum required prognosis accuracy

As explained in the introduction, these parameters will partially be based on cost factors. Statistical values as Mean Time Between Repair (MTBR) are not evaluated in this study, because the results will not have any impact on these parameters. For further information see e.g. (Saxena et al., 2008).

3. MODEL APPLICATION

In this section the results of an exemplary simulation model application are summarised. Due to IP reasons a detailed description of the maintenance process logic as well as particular process factors are not presented. Regarding the scenarios, introduced in section 2.5.2, the analysis represents data obtained from scenario 1. Results of the other scenarios are not presented in this paper due to IP reasons and model modifications not implemented yet.

3.1. Numerical Example

The conducted test run presents LRU-specific data for the ADIRU using the Lufthansa Airbus A320 fleet. The MRO data provides complete information for the ADIRU from the years 2010 to 2013. 294 exemplary replacement events at the homebase are generated. Since the LRU is not maintained periodically, all replacements are unscheduled.

According to redundancy requirements each aircraft has three ADIRUs. ADIRU 1 is classified as particularly critical (MEL RI A). Regarding the examined fleet, four modifications (part-numbers) of the ADIRU are currently in service (see Table 5).

Table 5. ADIRU-specific model input values.

Parameter	Value
number of events	294
installed ADIRUs per aircraft	3
MEL $RI_{ADIRU\ 1}$	A
MEL $RI_{ADIRU\ 2,3}$	C
different ADIRU modifications (PNs)	4

General simulation input parameters are defined in Table 6. The labour cost rate is an average value for different employee qualifications. In reality, different qualifications with varying cost rates apply. An ADIRU replacement does not require any extra materials, thus not creating additional material costs. Logistics are considered as overhead costs.

Table 6. Simulation input values.

Parameter	Value
$n_{Monte\ Carlo\ Runs}$	250
c_L	\$200 per man hour
$c_{Ops_{Delay}}$	\$82 per delay minute
$c_{O_{Logistics}}$	\$100 per component

3.2. Input Data Analysis

Analysing the preprocessed data input without any simulation, provides information about LRU-specific maintenance characteristics, made available through the event-wise data clustering. A target value, supposed to be reduced by PHM, is the component's NFF rate. The influence of particular event characteristics on the NFF ratio is illustrated in Table 7. The NFF rate provides information about the diagnosis accuracy. An ideal 100% accuracy is not realistic, since the aim of low-

ering risks of false positive statements (NFF), falsely assuming an LRU is defective, is opposed to the aim to reduce false negative statements, falsely assuming an LRU is functioning.

It is shown that 35% of all replacement events are classified as NFF. Replacements involving AOG times (7%) show a slightly higher NFF ratio. As expected, cost-intensive ground-times as AOGs mainly cause quick part removals even without exact findings. Subsequent NFF findings in the subsystem maintenance then often occur. However, the sample size is low in this case and a direct correlation cannot reliably be stated. Replacements, that were deferred in the past (22%), show a higher NFF ratio as well. This behaviour is not expected. A deferral should leave more time for troubleshooting, thus improving diagnosis quality resulting in less NFF cases. The ability of an aircraft to provide fault messages in real-time (RTS) (72% of the events) has no influence on the NFF ratio. Regarding the mounting location, the evaluation shows that the replacements are equally distributed over the different ADIRU positions. If the ADIRU 1 is affected, the NFF rate is lower. Since the ADIRU 1 is more critical (MEL RI A), this behaviour is contrary to the AOG results. On the other hand a higher components priority can lead to more precise troubleshooting, eventually creating less NFF events.

Table 7. NFF analysis w.r.t. event characteristics.

Event characteristic	n_{events}	n_{NFF}	$\frac{n_{NFF}}{n_{events}}[\%]$
1. all events	294	103	35.0
2.a) AOG	21	13	61.9
2.b) no AOG	273	90	32.9
3.a) deferred	66	40	60.6
3.b) non-deferred	228	63	27.6
4.a) with RTS	211	74	35.1
4.b) without RTS	83	29	34.9
5.a) ADIRU 1	94	13	13.8
5.b) ADIRU 2	91	46	50.5
5.c) ADIRU 3	109	44	40.4

By analysing LRU-specific delay characteristics the effects of a PHM system introduction can exactly be quantified. A delay analysis, concerning technically caused delays only, provides the results shown in Table 8. 20.4% of the events generated technically caused (primary) delays. The average delay duration is 18.1 minutes. Within subsequent flights further delays (secondary) were generated. Their accumulated average duration is 19.6 minutes. The results are relevant for the cost calculation in Section 3.3.3.

Analysing LRU data on an aircraft-based level provides information about correlations between events (see Figure 8). For three exemplary aircrafts it is shown that ADIRU replacements occur w.r.t. all mounting positions. Table 7 also illustrates the nearly equal distribution over all positions. A further analysis shows that within the period of examination 131 consecutive ADIRU replacements occur. Out of 131 events, 59 replacements (45%) occur at the same mounting position as the prior one, being slightly higher than the probability of an equally distributed behaviour (33% for 3 mounting positions). Probably not all replacements actually solved the root cause of the problem.

Table 8. Analysis of initial (primary) and subsequent (secondary) delays.

Delay type	n_{delay}	$\frac{n_{delay}}{n_{events}}[\%]$	$t_{O,mean}[min]$
primary delay	60	20.4	18.1
secondary delay	53	18.0	19.6

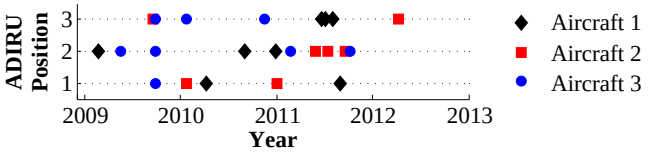


Figure 8. Aircaft-specific failure sequence analysis w.r.t. the ADIRU mounting positions.

3.3. Simulation Results

The following subsections deal with results obtained from the simulation.

3.3.1. Simulation States

The system states (see Table 3 and 4) of an exemplary event are illustrated in Figure 9. The subsystem state illustrates the point of time of failure ($t_{Simulation} = 0$) and the further processing. The failure rectification, starting after the aircraft has landed, is represented by $z_{Subsystem} = 0$.

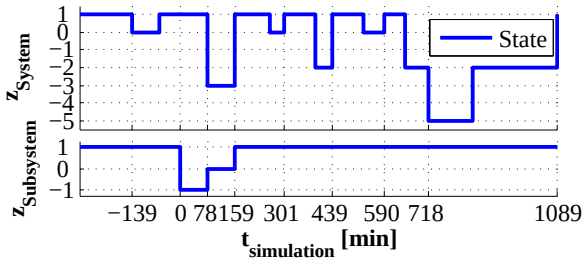


Figure 9. System and subsystem states of an exemplary event.

The plot primarily enables model validation by visualisation of the system states. It shows available maintenance times as well as generated delays and rectification process characteristics. z_{System} is a result of the flight plan and particular boundary conditions generated by maintenance actions. The effects on aircraft availability can be represented, if the entire flight operation is considered.

3.3.2. Time-based Analysis

Analysing processes w.r.t. temporal data, provides information about particularly time-consuming or delay-causing processes and modules. Concerning the ADIRU, the overall process time from failure identification to rectification is represented in Figure 10. The plot shows two distributions caused by different rectification procedures. If a failure occurs during flight operation and is classified as urgent, the rectification usually takes place at the ramp immediately (left distribution, short duration). If the complaint is deferred, the rectification is carried out in a hangar at the next planned plug (right distribution, long duration). This usually involves higher maintenance efforts, e.g. through detailed planning and repeated troubleshooting tasks and thus is more time-consuming. For the ADIRU the mean average is $t_{rectification} = 71.7min$.

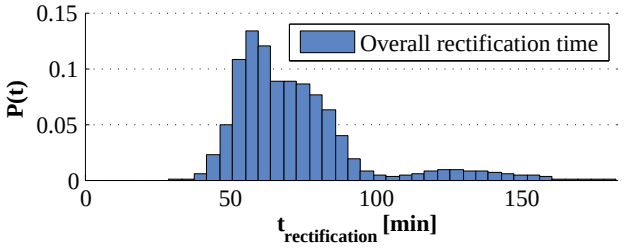


Figure 10. Overall processing time of ADIRU replacements.

Figure 11 shows ADIRU diagnosis process times. The mean average time is $t_{diagnosis} = 37.6min$. One aim of PHM is to consistently carry out system diagnosis prior to failures in order to reduce replacement durations. Since the average diagnosis time is almost half the average total rectification time, the effects on unscheduled groundtimes and delays are expected to be significant.

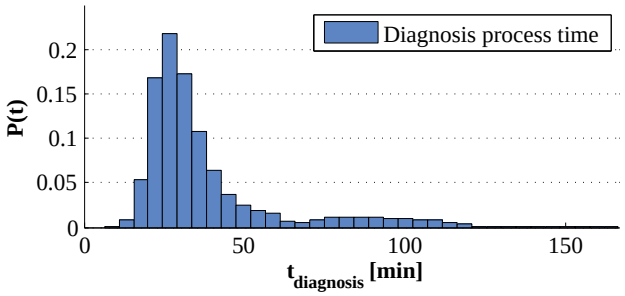


Figure 11. ADIRU diagnosis process time.

If a failure requires specific action, the TS creates an Action Order (AO), a detailed task manual. The completion of replacements with an AO requires more time in most cases, as confirmed by the results shown in Figure 12. Since events involving AOs can be classified as special case treatment, the use of PHM is expected to standardize the rectification and to reduce the number of AO processes.

Based on a detailed delay analysis w.r.t. process times, all

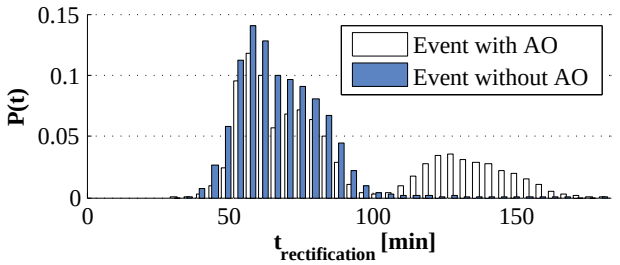


Figure 12. ADIRU processing time w.r.t. AO characteristics.

events are categorized into four classes. 79.6% of the ADIRU replacements do not generate any delays. 6.6% of the events cause delays, but could have been prevented, if the diagnosis processes were carried out prior to the unscheduled groundtime. 13.6% of the events generated delays that could only be prevented by planning the replacement into a prior groundtime. 0.2% of the events would always cause delays, because a unique ADIRU problem occured.

Based on the event characteristics of the second and third category (events with avoidable delays) the results shown in Figure 13 and 14 can be obtained.

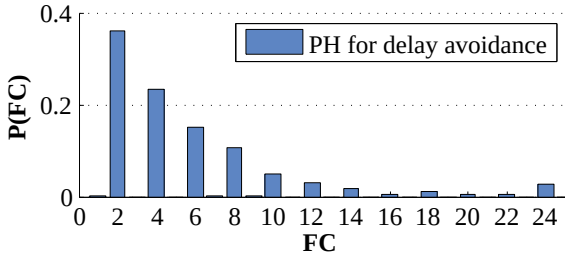


Figure 13. Required prognostic horizon for delay avoidance as a function of flight cycles.

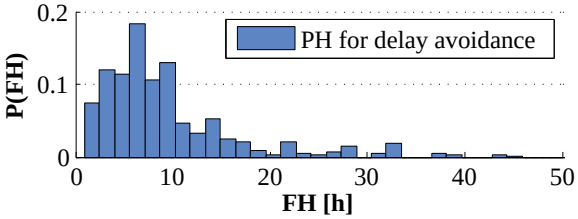


Figure 14. Required prognostic horizon for delay avoidance as a function of flight hours.

By means of the flight schedule and the calculated process times, the prior groundtime for every event, not generating a delay, can be identified. The necessary time-shift to that particular groudtime can be specified in terms of FC or FH, illustrated as a PDF. Since only replacements at the homebase are analysed in the first place, the FC analysis shows the expected behaviour that only every second flight is accounted for (groundtimes at the homebase). For instance, if an ideally working PHM system with a PH of 4 FC or 9 FH is used,

60% of the delays could have been avoided completely. Additionally the delays of other events could partially be reduced by scheduling them into more adequate groundtimes than the actual ones.

3.3.3. Cost-based Analysis

A cost-based analysis provides information about specific cost distributions. Table 9 gives an overview of the calculated ADIRU replacement costs. The average value for the annual costs as well as the lower and upper boundaries of the confidence interval (CI), including 95% of the values, are given. Due to deterministic input data, for logistics overhead costs no CI applies.

The average overall costs for ADIRU replacements sum up to $125,365 per year. One event generates average total costs of $1,706. The uncertainty is described by the given CI, ranging from $269 to $4,419. Two thirds of the costs of an ordinary replacement event are generated by MRO processes, one third by operational irregularities. The module-wise analysis shows that especially the maintenance modules and the logistics account for a large portion of the costs. A further analysis determines the costs of NFF events ($C_{Subsys.M._{NFF}}$) as a fraction of the subsystem maintenance costs. The subsystem maintenance process is the costliest process, due to the fact that all on-aircraft ADIRU tasks are performed quickly, whereas a detailed component maintenance - the ADIRU is a computer - is time-consuming. Furthermore, the costs of diagnosis tasks ($C_{Diagnosis}$) are analysed, being part of troubleshooting (C_{TS}), system maintenance ($C_{Sys.M.}$) and subsystem maintenance costs ($C_{Subsys.M.}$).

Table 9. ADIRU replacement cost analysis.

Cost type	**mean costs [per event]**	**95% CI [per event]** min - max	**mean costs [per year]**
C_{Event}	$1,706	$269 - $4,419	$125,365
C_{Ops}	$593	$0 - $2,291	$43,558
C_{Proc}	$1,113	$269 - $2,524	$81,807
C_{TS}	$35	$11 - $127	$2,597
$C_{Planning}$	$13	$8 - $22	$948
$C_{Sys.M.}$	$164	$112 - $207	$12,039
$C_{Subsys.M.}$	$801	$0.4 - $1,859	$58,873
$C_{Logistics}$	$100		$7,350
$C_{Subsys.M._{NFF}}$	$183	$26 - $432	$11,282
$C_{Logistics_{NFF}}$	$35		$2,573
$C_{Diagnosis}$	$125	$59 - $348	$9,212

Out of the listed costs only some are avoidable (eq. 4). These are delay costs C_{Ops}, costs of NFF events $C_{Subsys.M._{NFF}}$, logistics costs of NFF events $C_{Logistics_{NFF}}$ and costs of diagnosis processes $C_{Diagnosis}$. The avoidable, annual costs reach $C_{avoidable} = \$66,625$ or 53.1% of the average overall costs per year.

$$C_{avoidable} = C_{Ops} + C_{Subs._{NFF}} + C_{Log._{NFF}} + C_{Diag.} \quad (4)$$

3.3.4. Derivation of PHM Design Parameters

Based on the calculated operational and economic constraints, the benefit of particular PHM design parameters can be evaluated. Figure 15 shows the impact of different PHM system prognosis horizons, specified by the numbers of FH, and different prognosis accuracies on the costs of operational irregularities (C_{Ops}). An imperfect system is accounted for by a confidence value, representing a simplified accuracy. A confidence of 0.25 implies that 25% of the delay causing events could have been prevented by performing proactive maintenance. It is shown that an effective cost reduction requires a reliable prognosis (high confidence) as well as a sufficient PH (high number of FH). A full reduction of delay costs is not feasible because of few unavoidable major events within the evaluation period.

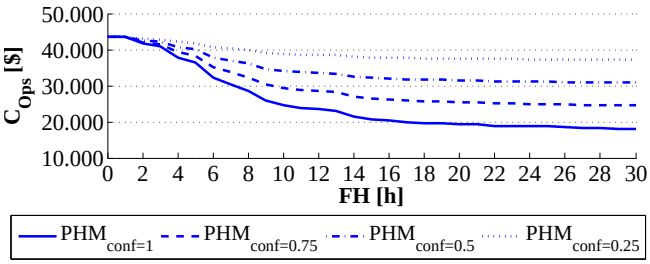


Figure 15. Impact of different inaccurate PHM systems with varying PH on costs of operational irregularities.

Some potential cost reductions are quantified in Table 10. The reductions for realistic PHM systems (confidence < 1, short PH) appear to be low. If the parameters of an exemplary PHM system are set to $PHM_{conf} = 0.5$ and $PH = 2$ FH, the potential savings reach $987 per year only. If investment costs of PHM systems are considered, the cost-benefit might turn out negative in the end.

Table 10. Impact on costs of operational irregularities w.r.t. prognosis accuracy and horizon.

PHM_{conf}	2 FH	5 FH	10 FH	20 FH
0.25	-$494	-$1,756	-$4,768	-$6,038
0.5	-$987	-$3,513	-$9,536	-$12,076
0.75	-$1,481	-$5,269	-$14,304	-$18,114
1.0	-$1,974	-$7,025	-$19,072	-$24,152

Besides the impact on delay costs, the influence on MRO process costs is evaluated as well. Table 11 gives an overview of potential savings concerning the aforementioned avoidable cost categories. It is assumed that the PHM system's confidence allows to avoid the calculated costs proportionally. For instance, a PHM system with 50% accuracy enables

the reduction of 50% avoidable costs, generating savings of \$11,534 per year in this case.

Table 11. Impact of different inaccurate PHM systems on avoidable MRO process costs.

PHM_{conf}	$C_{Sub._{NFF}}$	$C_{Log._{NFF}}$	$C_{Diag.}$	$\sum$
0.25	-\$2,821	-\$643	-\$2,303	-\$5,767
0.5	-\$5,641	-\$1,287	-\$4,606	-\$11,534
0.75	-\$8,462	-\$1,930	-\$6,909	-\$17,300
1.0	-\$11,282	-\$2,573	-\$9,212	-\$23,067

The overall savings potential is illustrated in Figure 16. It depends on accuracy and PH of the PHM system. Whereas the accuracy reduces costs in both categories, operational and MRO costs, a longer PH primarily allows to prevent more delays. So the effects on process costs only depend on the accuracy. For instance, a realistic PHM system for the ADIRU with 50% accuracy and $PH = 2FH$ reduces the avoidable costs to $C_{avoidable} = \$54,104$ per year, an annual reduction of \$12,521 or 18.8%. Since no investment costs are considered in this study, the savings potentials specify a boundary for reasonable PHM investment costs.

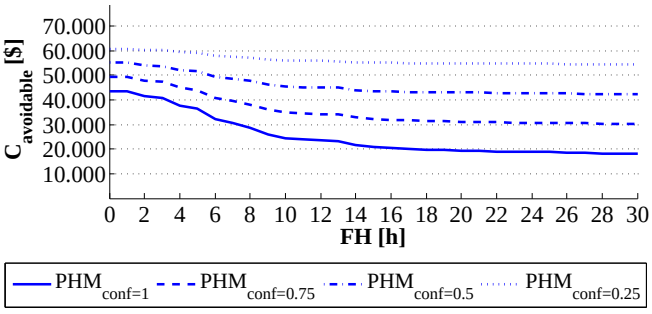


Figure 16. Impact of different inaccurate PHM systems with varying PH on avoidable costs.

Since no prognosis algorithm performance data is available for this study, the effects of correlations between PH and accuracy are not represented. It is assumed that a shorter PH will result in a higher prediction accuracy. By quantification of the exact correlations, the analysis quality and the conclusions could be described more detailed in the future.

3.4. Model Validation

The model validation is carried out by conducting plausibility checks. By comparing the simulated process sequences with the process analysis EPC model, the model logic is validated. A comparison of the simulated process time distributions to the input distributions verifies correct data usage. The system state diagram enables the validation of the interaction between flight operation and MRO processes. This way also the generation and recording of delay data can be confirmed. Further methods for model validation include Gantt charts for visualisation as well as process route marking for plausibility checking.

4. CONCLUSION AND OUTLOOK

This study presents a new approach for the assessment of PHM relevant components concerning avoidable costs of unscheduled events. The aim is to evaluate the characteristics of today's maintenance on LRU level and to derive design information for future PHM systems. Therefore, a DES model is built up in order to represent the MRO process logic using empirical maintenance data. After a data preprocessing is carried out, a Monte Carlo simulation enables the representation of uncertain parameters. Process times and costs of exemplary LRUs are calculated and analysed. Unique features of this study are the use of mostly deterministic data and the event-discrete approach. Both procedures allow to evaluate dependencies, causes and effects within replacements events.

The results of an exemplary LRU, the ADIRU, show a decent savings potential. Operational irregularities and non-productive MRO processes cause $\$66,625$ avoidable costs per year. A sensitivity analysis of the impact of imperfect PHM systems on the aforementioned costs reveals that the benefit largely depends on the prediction accuracy as well as the PH. Whereas the PH allows to facilitate planning processes and thereby reduces delay costs, a PHM system's accuracy mostly saves costs of non-productive MRO processes through improved diagnosis. Not considering PHM investment costs, a realistic PHM system allows to save approximately 20% of the annual costs for the entire fleet.

A final specification of a PHM system is enabled by a ROI calculation, considering avoidable as well as investment costs, and an analysis of the correlation between prognosis accuracy and horizon, providing prognosis algorithm performance characteristics. Future work will focus on the simulation of target state scenarios in order to evaluate the effects of different diagnosis and prognosis approaches in detail. Influential parameters will be considered by performing further sensitivity analysis. The analysis of a large number of LRUs will further improve the understanding.

It is assumed that there is a standardized LRU maintenance process and that the analysed LRUs show an observable wear behaviour. LRUs that do not meet these requirements, are not applicable for the simulation. Furthermore, the quality of the simulation results largely depends on the input data quality, as inaccurate or conflicting data degrades the conclusions.

ACKNOWLEDGMENT

This study was carried out as part of a research project of Lufthansa Technik AG. The authors would like to thank the project partners for the support and contributions to the research conducted in this paper.

ABBREVIATIONS

ADIRU	Air Data Inertial Reference Unit
AO	Action Order
AOG	Aircraft on Ground
ATA	Air Transport Association
CI	Confidence Interval
DES	Discrete Event Simulation
DMC	Direct Maintenance Costs
DOC	Direct Operating Costs
DR	Dispatch Reliability
EPC	Event-driven Process Chain
FC	Flight Cycle
FH	Flight Hour
IP	Intellectual Property
LRU	Line replacable Unit
MEL	Minimum Equipment List
MRO	Maintenance, Repair and Overhaul
MTBR	Mean Time Between Repair
NFF	No-Fault-Found
PDF	Probability Density Function
PH	Prognostic Horizon
PHM	Prognostics and Health Management
PN	Partnumber
RI	Rectification Interval
ROI	Return on Investment
RTS	Real-Time-Sending
RUL	Remaining Useful Life
TS	Troubleshooting

REFERENCES

Air Transport Association of America. (2012). *ATA iSpec 2200 - Information Standards for Aviation Maintenance.*

Bender, A., Pincombe, A. H., & Sherman, G. D. (2009). Effects of decay uncertainty in the prediction of life-cycle costing for large scale military capability projects. In *18th World IMACS/MODSIM Congress, Cairns, Australia.* (Vol. 1).

Beynon-Davies, P. (2004). *Database systems* (3rd ed.). Basingstoke: Macmillan.

Cook, A. J., Tanner, G., & Anderson, S. (2004). *Evaluating the true cost to airlines of one minute of airborne or ground delay: final report.* Eurocontrol.

Eurocontrol. (2010). Eurocontrol Performance Review Commission: An Assessment of Air Traffic Management in Europe during the Calendar Year 2009.

Eurocontrol. (2011). Planning for Delay: Influence of Flight Scheduling on Airline Punctuality. *EUROCONTROL Trends in Air Traffic*(Volume 7).

Eurocontrol. (2012). Eurocontrol Central Office for Delay Analysis: CODA Digest - Delays to Air Transport in Europe November 2011.

Feldman, K., Jazouli, T., & Sandborn, P. A. (2009). A methodology for determining the return on investment associated with prognostics and health management. *IEEE Transactions on Reliability*, *58*(2), 305–316.

Fritzsche, R., & Lasch, R. (2012). An Integrated Logistics Model of Spare Parts Maintenance Planning within the Aviation Industry. *Proceedings of world academy of science, engineering and technology*, *Volume 68*.

Fromm, H. B. (2009). *Bewertung innovativer Instandhaltungsszenarien in den frühen Phasen des Innovationsprozesses in der Luftfahrt*. Aachen: Shaker.

Gray, M. A. (2007). Discrete event simulation: A review of SimEvents. *Computing in Science & Engineering*, *9*(6), 62–66.

Hölzel, N. B., Schilling, T., Neuheuser, T., Gollnick, V., & Lufthansa Technik AG. (2012). System Analysis of Prognostics and Health Management Systems for Future Transport Aircraft. In *28th International Congress of the Aeronautical Sciences (ICAS), Brisbane, Australia.*

Knotts, R. M. (1999). Civil aircraft maintenance and support Fault diagnosis from a business perspective. *Journal of Quality in Maintenance Engineering*, *5*(4), 335–348.

Kohn, W. (2005). *Statistik: Datenanalyse und Wahrscheinlichkeitsrechnung*. Berlin: Springer.

Linser, A. (2005). *Performance Measurement in der Flugzeuginstandhaltung* (Unpublished doctoral dissertation). Hochschule für Wirtschafts-, Rechts-und Sozialwissenschaften, St. Gallen.

Rodrigues, R. S., Balestrassi, P. P., Paiva, A. P., Garcia-Diaz, A., & Pontes, F. J. (2012). Aircraft interior failure pattern recognition utilizing text mining and neural networks. *Journal of Intelligent Information Systems*, *38*(3), 741–766.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *2008 International Conference on Prognostics and Health Management* (pp. 1–17). IEEE.

Sisk, L. (1993). Analysing the value of maintenance in terms of despatch reliability. *Cost Effectiveness Maintenance Conference*.

Sun, B., Shengkui Zeng, Kang, R., & Pecht, M. (2010). Benefits analysis of prognostics in systems. In *2010 Prognostics and System Health Management Conference* (pp. 1–8). IEEE.

A Framework for Model-Based Diagnostics and Prognostics of Switched-Mode Power Supplies

Honglei Li[1], Xuerong Ye[2], Cen Chen[3], George Vachtsevanos[4]

[1, 4] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA*
honglei.li@gatech.edu
gjv@ece.gatech.edu

[2, 3] *School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin, 150001, China*
xuelai1981@163.com
chencen65061700@sina.com

ABSTRACT

With electrical power supplies playing an important role in the operation of aircraft systems and sub-systems, flight and ground crews need health state awareness and prediction tools that accurately diagnose faults, predict failures, and project remaining life of these onboard power supplies. Among onboard power supplies, switch-mode power supplies are commonly used where their weight, size, and efficiency make them preferable to conventional transformer-based power supplies. In this paper, we present a framework of diagnostics and prognostics methodology based on an equivalent circuit system simulation model developed from a commercially available switch-mode power supply, and empirical component degradation models. In industrial applications, case-specified modifications can be made according to specific experimental or service conditions of different commercial products. First, the developed simulation model is validated through experimental testing. Then, a series of data are collected from simulation to build the baseline and fault databases under a fixed load profile. Next, promising features are extracted from sensed parameters, and further data analysis are conducted to estimate the current health condition and to predict the remaining useful life of the target system. Some highlights of the work are included but not only limited to the following aspects: first, the methodology is based on electronic system simulation instead of traditional accelerated testing by employing a high-fidelity system simulation model and empirical critical component degradation models; second, efforts are made in this preliminary work to adapt proven prognostics and health management techniques from machinery to electronic health management, with the goal of expanding the realm of electronic diagnostics and prognostics.

Honglei Li et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Electronic systems such as electronic controls, onboard computers, communications, navigation and radar perform many critical functions in onboard military and commercial aircrafts. All of these systems depend on electrical power supplies for direct current power at a constant voltage to drive solid-state electronics. With these power supplies playing an important role in the operation of aircraft systems and sub-systems, flight and ground crews need health state awareness and prediction tools that diagnose faults accurately, predict failures, and project remaining useful life (RUL) of these components. Among various electrical power supplies, switch-mode power supplies (SMPS's) are commonly used in onboard aircrafts where their weight, size, and efficiency make them preferable to conventional transformer-based power supplies.

Traditional reliability practices applied in electronics are limited to reliability analysis based on historic reliability statistics and ageing models/factors of population-specific components from commonly accepted resources. Few efforts target at developing high fidelity models for specific electronic systems. On the other hand, many current prognostic and health management (PHM) practices rely on extensive highly accelerated life testing (HALT) to obtain degradation/failure data or models, which may substantially increase product life cycle costing (Brown, D. W., Kalgren, P. W., & Roemer, M. J., 2007). To address the need of developing higher fidelity models and reducing the life cycle costing, this paper proposes the use of a model-based diagnostics and prognostics approach for specific electronic systems, integrating reliability statistics, domain expertise, with experimental testing verification. More specifically, in this paper, the efforts are made to develop processes that adapt proven PHM concepts from machinery health management to electronic systems with the utilization of an integrated simulation model combining two empirical models in the application of SMPS: a circuit-based SMPS simulation model and the components' degradation models

developed based on domain expertise and validated via experimental testing.

A schematic diagram of the proposed model-based SMPS diagnostics and prognostics methodology is as shown in Figure 1. First, a high-fidelity SMPS system simulation model is established and validated via actual system testing under a fixed load profile. Single critical component is selected with the consideration of both the reliability statistics and the specific application. Then, in the fault diagnostics module, simulated data are generated to build baseline and fault databases under the same load profile. Probability of detection (POD) is selected and calculated over time for the purpose of fault detection and the trigger of failure prognosis. In the failure prognostics module, system degradation model is developed and then a model-based particle filter routine is adopted to estimate the model parameters and finally, predict RULs. Note that, all models, experimental results and analysis discussed in this paper pertain to a commercial-available SMPS as shown in Figure 2. The target SMPS system is a constant current source with the output current of 700mA ±15mA.

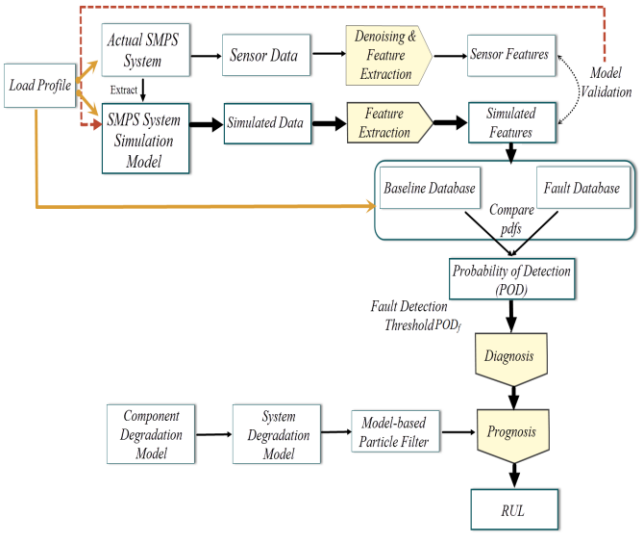


Figure 1. Systematic diagram of the proposed methodology.

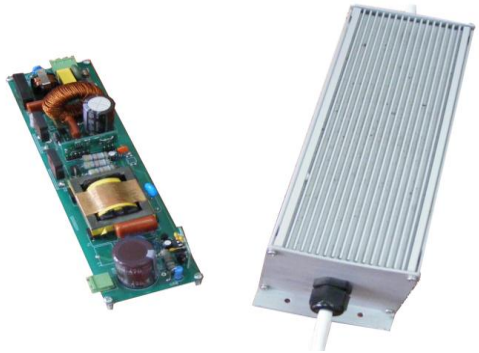


Figure 2. The SMPS commercial product.

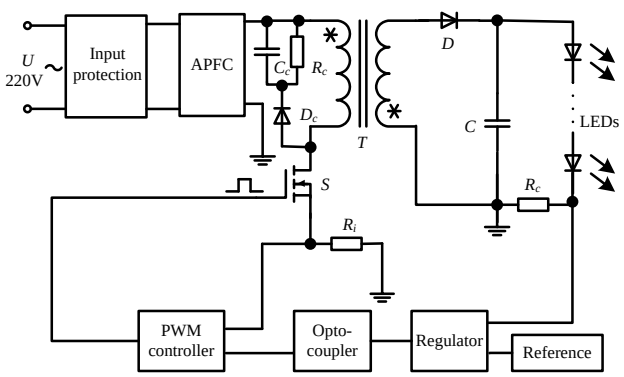
2. MODELING METHODOLOGY

In this section, the above-mentioned two types of empirical models are introduced: the circuit-based SMPS system simulation model and the critical components' degradation models, from which an integrated simulation model is generated to serve in the framework of diagnostics and prognostics to be introduced in Section 3.

2.1 SMPS System Modeling

2.1.1 Model Development

A circuit-based simulation model for the target SMPS system was developed using software PSpice. OrCAD PSpice is a *Simulation Program with Integrated Circuit Emphasis* (SPICE) analog circuit and digital logic simulation and analysis program, which is widely used in academia and industry. First, equivalent circuit models were built for individual components, for example, transformers. Then, all component models were integrated to build the SMPS system circuit model as shown in Figure 3. The whole SMPS consists of the input protecting circuit, Active Power Factor Corrector (APFC), opto-isolator, comparing regulator and other parts. The loads are 44 LEDs in serial connection, as shown in Figure 3.



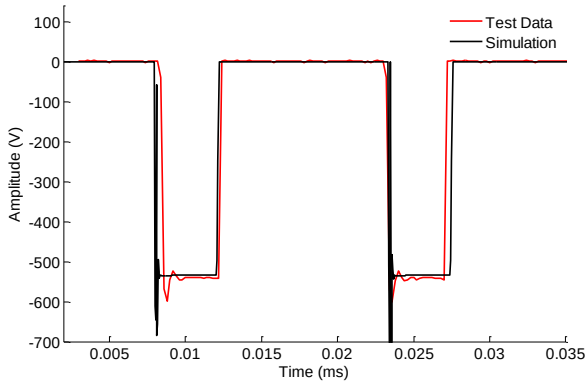
C in this circuit is the aluminum electrolytic capacitor that will age.

Figure 3. SMPS model schematic diagram.

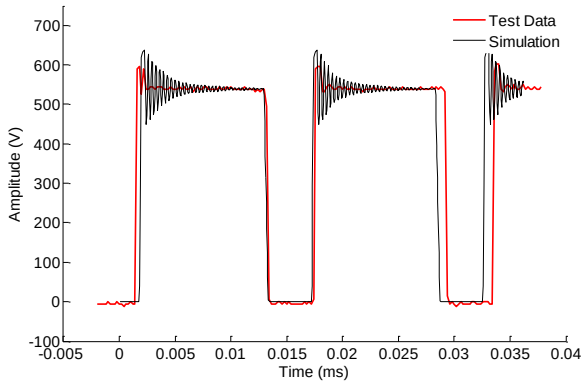
2.1.2 Model Validation

Model validation is crucial to the high-fidelity simulation model establishment. To validate the established model, critical model parameters are usually compared to the corresponding experimental outputs from selected testing points. In this case, several comparison parameters were selected such as MOSFET drive signals (i.e., V_{gs}, V_{ds}) and diode D voltage. The MOSFET drive signal waveforms from the model and the experiment are as shown in Figure 4 as an example. As indicated in Figure 4, the model performances generally match with the experimental results, and the simulation model is validated. Note that in Figure 4, according to the authors' domain experience, high-frequency oscillation observed at the simulated waveform

changing edges could be attributed to the simulation algorithm design, and the small discrepancy between simulation and testing values could be due to the testing temperature variation and/or the actual system's Pulse-Width Modulation (PWM) chip output voltage variation.



(a)



(b)

Figure 4. Simulation and experimental test waveforms of MOSFET drive signals: (a) V_{gs}, (b)V_{ds}.

2.2 SMPS Degradation Modeling

It has been established in several works (Zhang, Kang, Luo and Pecht, 2009; Goodman, Hofmeister, and Judkins, 2007) that component degradation, especially the critical components' degradation, is the prime contributor to SMPS system degradation and eventually functional failure. Thus, it is essential to identify the critical components and study their degradation progression trends. Here our interest is to study the target SMPS system's soft failure induced by system's functional degradation under a fixed load profile, and our hypothesis is that the SMPS system's degradation is only caused by the single critical component's degradation. Thus, the system assumes the same degradation model as the critical component.

2.2.1 Critical Component Identification

Previous reliability studies of typical SMPS components have shown that the majority of failures may be attributed to a list of critical components such as *metal-oxide semiconductor field-effect transistors* (MOSFETs), aluminum electrolytic capacitors and silicon power rectifier diodes (Li, D., & Li, X., 2012). The failures of those components correspond to approximately 80% of the total failures. In this work, in addition to component reliability studies, a failure mode and effects analysis (FMECA) was also conducted to generate a list of critical components for this specific commercial SMPS. In this paper, for the purpose of illustration of methodology, aluminum electrolytic capacitor and feedback resistor are selected for single critical component degradation study.

2.2.2 Critical Component Degradation Modeling

System/component degradation modeling is tightly connected with the usage, environmental and operational conditions, or, the corresponding load profile *U* composed of critical stress factors. It is recommended in practice to integrate the stress factor influence into the degradation modeling. However, studying the fault progression as a function of varied load profiles is beyond the scope of this paper. Thus, here, we fix the SMPS load profile including three stress factors: input voltage, load resistance and temperature. For the choice of modeling approach, we adopt the feature-based modeling, as the degradation of electronic components usually reflects in their performance parameters' drifting from the nominal values.

a) Aluminum Electrolytic Capacitor Degradation

Aluminum electrolytic capacitors are known for their comparatively low reliability, and due to their criticality in SMPS systems they are a good candidate to study their degradation modeling and its contribution to system's failure. The performance of those components depends on the anode metal oxide film. With the thickening of anodic metal oxide film, the equivalent series resistance (ESR) increases and its capacitance decreases, while hydrogen produced from the cathode reaction accelerates the evaporation of electrolyte, which causes aluminum electrolytic capacitors' degradation.

The equivalent circuit model of the aluminum electrolytic capacitor in this application is as shown in Figure 5. In Figure 5, C_7 and C_{11} represent capacity values; R_{39} and R_{43} represent ESR values.

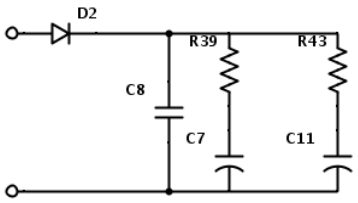


Figure 5. Aluminum electrolytic capacitor equivalent circuit model in PSpice.

Given a fixed operational temperature, the capacitor degradation rate is constant. The capacity and ESR values change as the aluminum electrolytic capacitor degrades, as expressed in Equations (1) and (2):

$$ESR(\mathrm{t}) = a_1 + \log(b_1 \cdot t + 1) \quad (1)$$

$$C(\mathrm{t}) = a_2 - b_2 \cdot t \quad (2)$$

where $a_1 =$ 0.3 Ω, a_2 =220 μF, $\mathrm{b}_1 = 7 \times 10^{-5}$, $\mathrm{b}_2 = 3 \times 10^{-5}$. The degradation model parameter values are empirically selected.

b) Feedback Resistor Degradation

In an SMPS system, the feedback circuit monitors the output voltage and compares it with a reference voltage. In the feedback loop, the degradation of feedback resistor plays a vital role in SMPS's reliability. Theoretically, with the reference voltage unchanged, an increase of feedback resistance will lead to a decrease of SMPS output current as indicated in Equation (3):

$$\mathrm{I} = I_N \times \frac{R_N}{R_a} \quad (3)$$

where $\mathrm{I_N}$ and $\mathrm{R_N}$ are SMPS average output current and feedback resistance under healthy condition, and R_a is the degraded feedback resistance. In this SMPS module, the feedback resistor is composed of two resistors in parallel. The empirical degradation models are as shown as follows:

$$R_{a1} = 3.9 + 7.8e^{-6} \cdot t$$

$$R_{a2} = 3.9 + 9.4e^{-6} \cdot t. \quad (4)$$

3. METHODOLOGY FOR MODEL-BASED DIAGNOSTICS AND PROGNOSTICS

In the field of PHM, fault diagnostics and failure prognostics techniques are usually classified according to the way that data is used to describe the behavior of the system: data-driven or model-based approaches. When the domain expertise is available to build a reliable degradation model of the monitored system, model-based diagnostics and prognostics approaches are preferable than the data-driven techniques. Figure 6 shows the systematic diagram for the proposed framework of model-based diagnostics and prognostics with Particle Filter (PF). In this case, the real-time data comes from the simulation model.

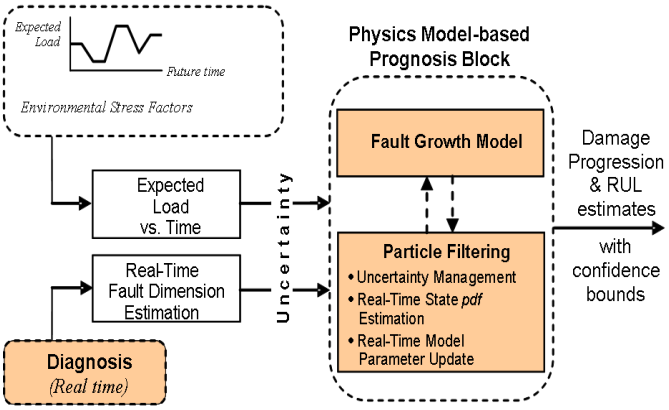


Figure 6. Model-based diagnostics and prognostics diagram.

3.1 Model-Based Diagnostics Module

A fault diagnostics module involves the tasks of fault detection and isolation, and identification (FDI). In general, this procedure may be interpreted as the fusion and utilization of the information present in a feature vector (measurements), with the objective of determining the operation states (i.e., being healthy or fault presence) of a system and the causes for deviations from particularly desired behavioral patterns.

In the model-based diagnostics framework, at any given instant of time, it provides a probability distribution function (PDF) estimate for meaningful physical variables in the system. In this case, simulation measurements at every time instant were collected from the integrated simulation model as introduced previously, and PDFs were generated from corresponding measurement histograms. Then, hypothesis testing through calculating current and baseline PDFs is used to generate fault alarms, and other statistical analysis tools may be used to extract additional information about the detection and diagnostic results. For example, in this case, POD is defined as below:

POD = 1 – Type II error.

Based on the calculated PODs from simulation, a fault detection threshold is set up in terms of POD. An illustrative example of fault detection confidence derived from type II statistical hypothesis testing with an example fault detection threshold is as shown in Figure 7. An illustration of fault progression with regard to the comparison of current and the baseline PDFs are as shown in Figure 8.

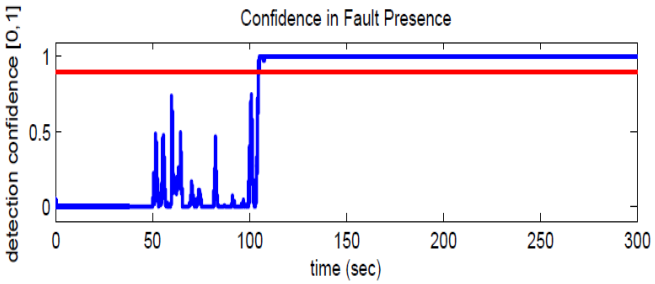
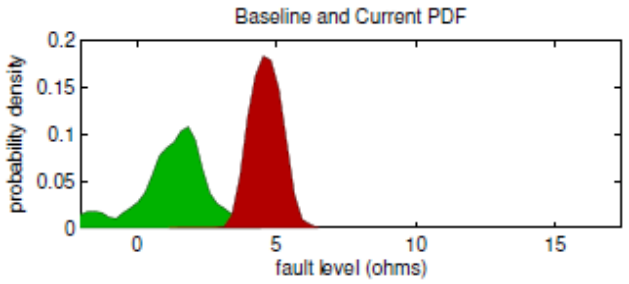
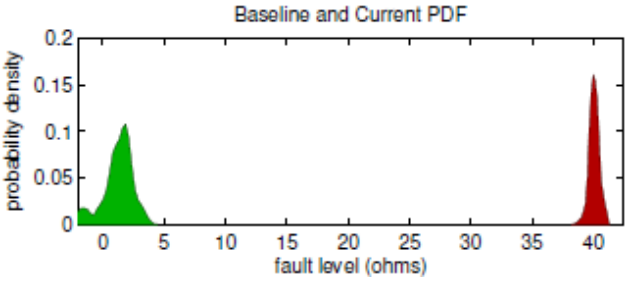


Figure 7. Estimator confidence metric derived from type II statistical hypothesis testing.



(a)



(b)

Figure 8. Baseline (left) and estimated (right) PDFs of (a) the mild and (b) the severe fault levels.

3.2 Model-Based Prognostics Module

A health-based failure prognostics module is usually triggered after the fault is detected, and the major task is to estimate RUL of the target system/component. In the process of model-based prognostics, the degradation model is expressed as a function of given load profile *U*, time *t*, and model parameters to be estimated Θ, or, mathematically,

$$\xi = \xi(t, \theta, U). \quad (5)$$

Note that Load profile *U* includes the contribution from the system external inputs and different stress factors as introduced before. The model parameters are estimated by integrating the degradation model with the observed health data. The RUL is calculated based on estimated model parameters.

In this paper, we realize the model-based prognostics in the PF framework. The methodology takes advantage of the empirical fault/degradation model, and a nonlinear process, a Bayesian estimation method using PF and real-time measurements. A merit of using PF for model-based prognostics is it combines RUL prediction and model estimation. Prognosis is achieved by performing two sequential steps, prediction and filtering. Prediction uses both the knowledge of the previous state estimate and the process model to generate the a priori state PDF estimate for the next time instant, or mathematically,

$$p(x_t|y_{1:t}) = \int p(x_t|x_{t-1})\, p(x_{0:t-1}|x_{1:t-1}) dx_{0:t-1}. \quad (6)$$

Unfortunately, this expression does not have an analytical solution in most cases. Instead, Sequential Monte Carlo (SMC) algorithms, or PF, are used to numerically solve this equation in real-time through the use of efficient sampling strategies. PF approximates the state pdf using samples or "particles" having associated discrete probability masses ("weights"), as expressed in Equation (7),

$$p(x_t|y_{1:t}) \approx \widetilde{w_t}(x^i_{0:t}) \cdot \delta(x_{0:t} - x^i_{0:t}) dx_{0:t-1}, \quad (7)$$

where $x^i_{0:t}$ is the state trajectory and $y_{1:t}$ are the measurements up to time t. The simplest implementation of this algorithm, the Sequential Importance Re-sampling (SIR) particle filter, updates the weights using the likelihood of y_t as

$$w_t = w_{t-1} \cdot p(y_t|x_t). \quad (8)$$

Long-term predictions are used to estimate the probability of failure in a system given a hazard zone that is defined via a probability density function with lower and upper bounds for the domain of the random variable, denoted as H_{lb} and H_{ub}, respectively. The probability of failure at any future time instant is estimated by combining both the weights $w^{(i)}_{t+k}$ of predicted trajectories and specifications for the hazard zone through the application of the Law of Total Probabilities. The resulting RUL PDF, where t_{RUL} refers to RUL, provides the basis for the generation of confidence intervals and expectations for prognosis,

$$\hat{p}_{t_{RUL}} = \sum_{i=1}^{n} p(Failure|X = \hat{x}^{(i)}_{t_{RUL}}, H_{lb}, H_{ub}) \cdot \quad (9)$$

In this case, we use a predetermined failure threshold instead of a hazard zone for the illustration of methodology.

4. RESULTS

In the SMPS simulated degradation process, we fixed a load profile of temperature $T = 25℃$, input voltage $V = 400V$, load resistance $Z = 220\Omega$, ran the integrated simulation model and monitored 10 output parameters: output current, voltage ripple, capacitance current ripple, capacitance voltage, transformer consumption, MOSFET consumption, MOSFET voltage, diode reverse voltage, and 47K resistance consumption.

4.1 Case Study: Aluminum Electrolyte Capacitor

4.1.1 Model-based Diagnostics

In the above-mentioned 10 output parameters, the amplitude of the output voltage ripple (VR) was substantially influenced by the degradation of aluminum electrolyte capacity. Therefore, VR amplitude was selected as a raw feature for further processing. In one cycle of SMPS degradation simulation, we collected 13 baseline and fault VR datasets with the time step of "thousand-hours", i.e., at $t = 0h, 1000h, \ldots, 12000h$. At every time step, Gaussian noise $\omega_k \sim N(0,0.01)$ was added to every VR measurement to represent uncertainty introduced by measurement noise, and 60 measurements of VRs were collected with an example as shown in Figure 9. Based on the measurement, the histograms were computed and the histogram of every faulty dataset was compared to the one of the baseline dataset with an example as shown in Figure 10, and the PDF was computed from the corresponding histogram. Then POD was calculated and recorded as shown in Table 2. Note that, in this case, we fixed the false alarm rate (Type I error) at 5% and monitored POD change as fault evolves. Recall that POD = 1 – Type II error. Figure 10 and Table 2 both show that the POD values increased as the SMPS degraded over time. Here, we chose POD=95% as the SMPS fault detection threshold to trigger our prognosis module. As indicated in Table 2, based on the given fault detection threshold, the first 8 datasets (i.e., t=0h, 1000h, …, 7000h) was regarded as the training data sets, while the last 5 (i.e., t=8000h, …, 12,000h) as the testing datasets.

Table 2. POD between the faulty and the baseline datasets.

t (kh)	1	…	4	5	6	7	8	9	…	12
POD	0	…	0	0.018	0.334	0.769	0.994	1	…	1

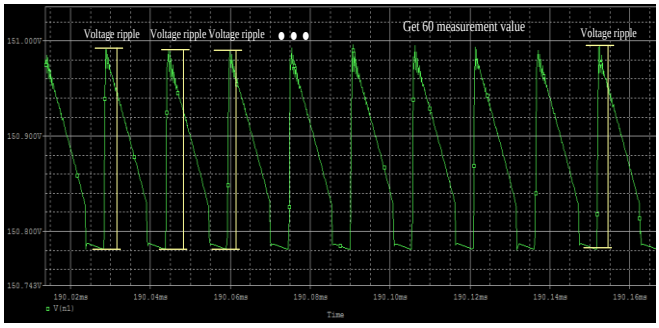


Figure 9. VR baseline (t=0h) measurements in PSpice.

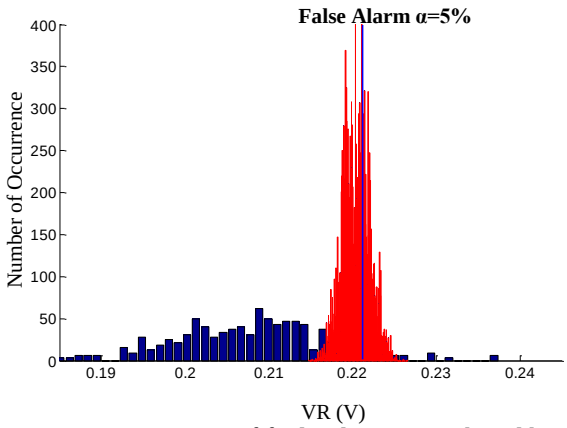


Figure 10. Comparison of faulty data at 6000h and baseline histograms.

4.1.2 Model-based Prognostics with Particle Filter

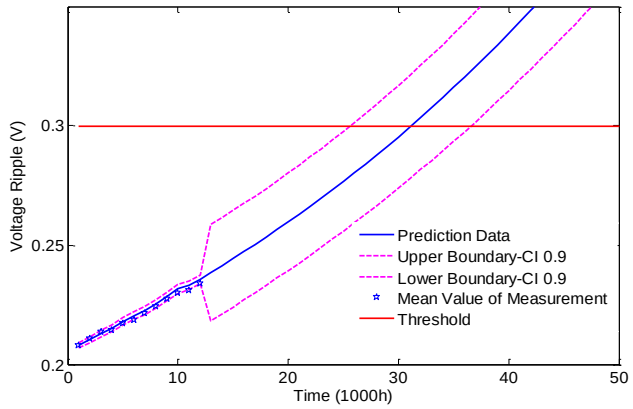
Once the fault detection threshold (i.e., POD = 95%) was reached, the SMPS RUL prognosis routine was triggered. An empirical degradation model is expressed by an exponential growth model as

$$x = a \cdot \exp(bt) + c, \tag{10}$$

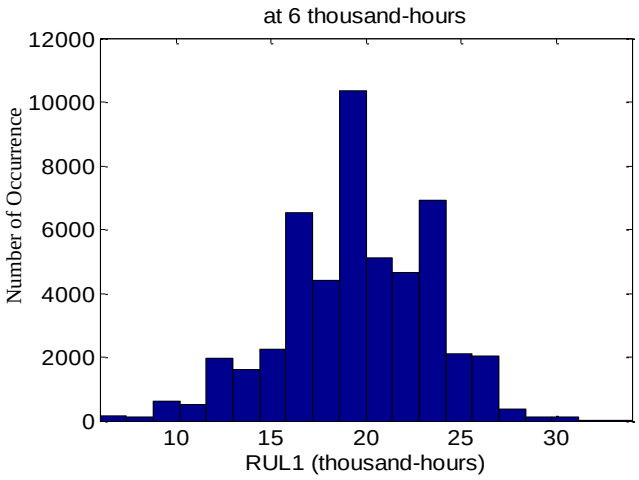
where x is VR, t is time, and a, b, c are unknown model parameters. The above SMPS degradation model can be rewritten in an iterative form of

$$x_{t_k} = \exp(b\Delta t)\left(x_{t_{k-1}} - c\right) + c. \tag{11}$$

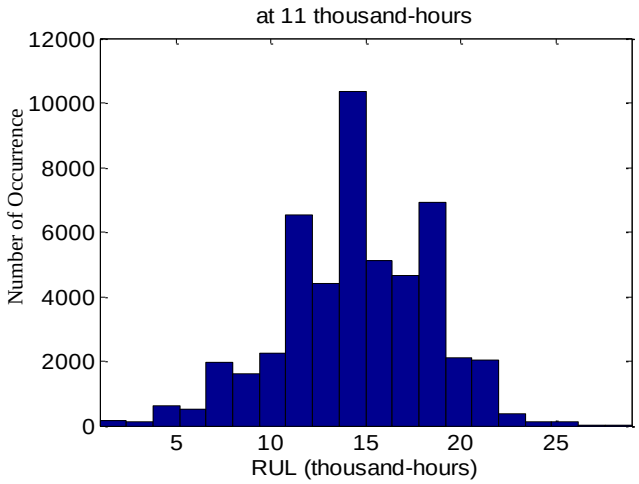
Both the model parameters and the RULs were estimated using PF. Here empirically we set the SMPS performance-based failure threshold as VR=0.3. The prediction diagram results in the form of probability are shown in Figure 11 (a). Figure 11 (b) and (c) show the RUL predictions at arbitrary cycles of 6,000h and 1,1000h respectively, in the form of distribution along with the 90% confidence interval (CI). As indicated in Figure 11 (b) and (c), the probabilistic RUL prediction was updated and the prediction accuracy improved over time.



(a)



(b)

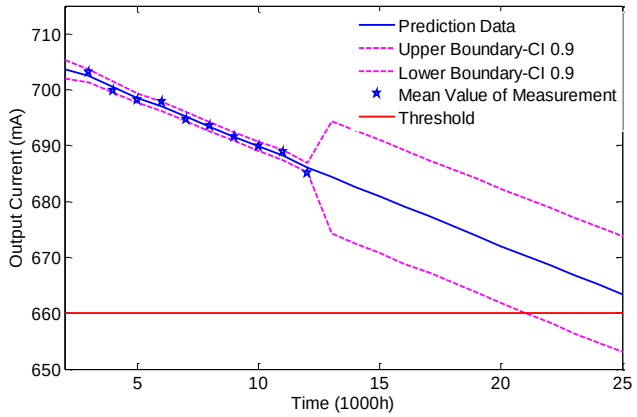


(c)

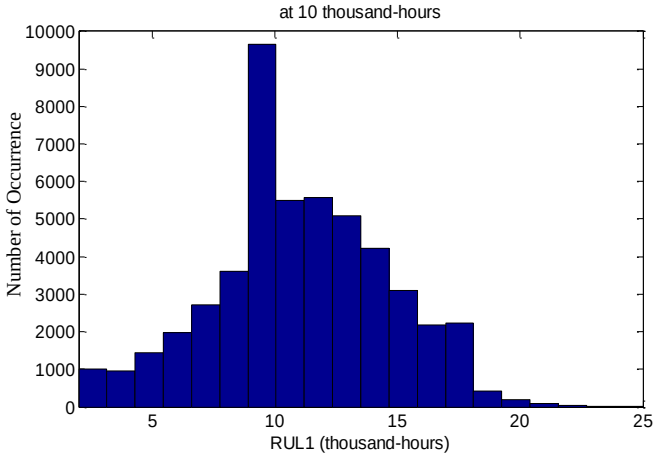
Figure 11. SMPS prognostics results in the case of aluminum electrolyte capacitor degradation: (a) prognosis module diagram results, (b) RUL pdf prediction at t=6,000 h, and (c) RUL pdf prediction at t=11,000 h.

4.2 Case Study: Feedback Resistor Degradation

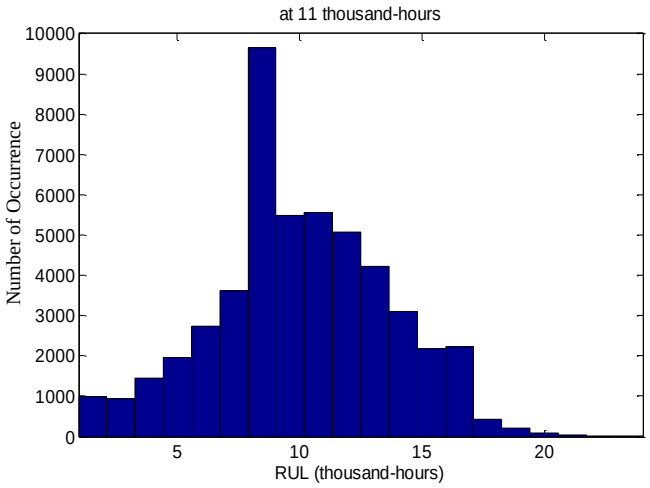
The above-mentioned methodology is also adapted to the case of feedback resistor degradation diagnostics and failure prognostics. RUL results are illustrated in Figure 12. As indicated in Figure 12, the output current decreased as the feedback resistor degraded over time.



(a)



(b)



(c)

Figure 12. SMPS prognostics results in the case of feedback resistor degradation. (a) prognosis module diagram results, (b) RUL pdf prediction at t=10,000 h, and (c) RUL pdf prediction at t=11,000 h.

5. CONCLUSIONS

This paper introduces a novel framework of a model-based SMPS fault diagnostics and failure prognostics methodology, which leverages the knowledge of the component physics and degradation physics to assess the health status, diagnose faulty conditions and predict RULs. The methodology is based on electronic system simulation by employing a high-fidelity system simulation model and empirical critical component degradation models. General procedures and simulation results are presented in two case studies of critical component degradation. Although the discussion is limited in the scope of a specific simulated model from a commercially available SMPS product, the methodology can be extended to other SMPS systems with related adjustment of the simulation model and the component degradation models based on corresponding system test results and the knowledge of critical component ageing behaviors. Future work is needed to study other cases for single critical component degradation, to study the scenario when multiple faults are injected simultaneously (i.e., multiple component degradation), to study the impact of varied loads on the RUL predictions, and to explore the damage accumulation degradation modeling approach in addition to the feature-based modeling approach as adopted in this paper.

REFERENCES

Brown, D. W., Kalgren, P. W., & Roemer, M. J. (2007). Electronic Prognostics – A Case Study Using Switched-Mode Power Supplies (SMPS). *IEEE Instrumentation and Measurement Magazine*, vol. 10(4), pp. 20-26.

Brown, D. W., Abbas, M., Ginart, A., Ali, I.N., Kalgren, P. W., & Vachtsevanos, G. J. (2012). Turn-Off Time as an Early Indicator of Insulated Gate Bipolar Transistor Latch-up, *IEEE Transactions on Power Electronics*, vol. 27(2), pp. 479-89.

Goodman, D., Hofmeister, J., & Judkins, J. (2007). Electronic Prognostics for Switched Mode Power Supplies. *Microelectronics Reliability*, vol. 47(12), pp. 1902-6.

Li, D., & Li, X. (2012). Study of Degradation in Switching Mode Power Supply Based on the Theory of PoF. *International Conference on Computer Science and Service System*, Aug. 2012, pp. 1976-1980.

Luo, J., Pattipati, K.R., Qiao, L., & Chigusa, S., (2008). Model-based Prognostic Techniques Applied to a Suspension System. *IEEE Transactions on System, Man and Cybernetics*, vol. 38(5), pp. 1156-1168.

MIL-HDBK-217F, Reliability Prediction of Electronic Equipment (1991). Department of Defense. Washington D.C..

Orchard, M. E., & Vachtsevanos, G. J., (2007). A Particle Filtering Approach for On-line Failure Prognosis in A Planetary Carrier Plate. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7(4), pp. 221-227.

Orsagh, R., Brown, D. W., Roemer, M., Dabvey, T., & Hess, A. (2005). Prognostic Health Management for Avionics System Power Supplies. *2005 IEEE Aerospace Conference* (IEEE Cat. No. 05TH8788), pp. 3585-91.

Zhang, H., Kang, R., Luo, M. & Pecht, M. (2009). Precursor Parameter Identification for Power Supply Prognostics and Health Management. *IEEE 8th International Conference on Reliability, Maintainability and Safety*, Jul. 2009, pp. 883–887.

Zhai, G., Zhou, Y., & Ye, X. (2013). A Tolerance Design Method for Electronic Circuits Based on Performance Degradation. *Quality and Reliability Engineering International*. DOI: 10.1002/qre.1621.

Anomaly Detection Techniques for the Condition Monitoring of Tidal Turbines

Grant S. Galloway[1], Victoria M. Catterson[2], Craig Love[3] and Andrew Robb[4]

[1, 2] *Institute for Energy and Environment, Dept. EEE, University of Strathclyde, Glasgow, G1 1XW, UK*

grant.galloway.2013@uni.strath.ac.uk
v.m.catterson@strath.ac.uk

[3, 4] *Andritz Hydro Hammerfest, Glasgow, G52 4RU, UK*

ABSTRACT

Harnessing the power of currents from the sea bed, tidal power has great potential to provide a means of renewable energy generation more predictable than similar technologies such as wind power. However, the nature of the operating environment provides challenges, with maintenance requiring a lift operation to gain access to the turbine above water. Failures of system components can therefore result in prolonged periods of downtime while repairs are completed on the surface, removing the system's ability to produce electricity and damaging revenues. The utilization of effective condition monitoring systems can therefore prove particularly beneficial to this industry.

This paper explores the use of the CRISP-DM data mining process model for identifying key trends within turbine sensor data, to define the expected response of a tidal turbine. Condition data from an operational 1 MW turbine, installed off the coast of Orkney, Scotland, was used for this study. The effectiveness of modeling techniques, including curve fitting, Gaussian mixture modeling, and density estimation are explored, using tidal turbine data in the absence of faults. The paper shows how these models can be used for anomaly detection of live turbine data, with anomalies indicating the possible onset of a fault within the system.

1. INTRODUCTION

Tidal power has great potential worldwide to be a major contributing source of renewable energy. It is a European target for 20% of energy generation to come from renewable resources by 2020, as stated in the European Union Committee 27th Report of Session 2007-08. Within the UK alone, tidal stream generation could potentially supply over 4 TWh per year within the next 5 to 10 years, with the potential to reach up to 94 TWh per year with an installed capacity of 36 GW (King & Tryfonas, 2009), around 26% of the total electricity generated within the UK in 2013 (UK Government electricity statistics). It is therefore clear that tidal energy has the potential to provide a major contribution to renewable sources of energy.

However, tidal power technology is in its infancy, and no clear tidal turbine design has emerged as an industry standard for extracting energy from tidal flow. The state of the art in turbine design includes many horizontal and vertical axis solutions, some with major structural and operational variations (Aly & El-Hawary, 2011). However, a common focus is the horizontal axis design, holding many similarities with a standard wind turbine.

Maintenance on tidal turbines requires a lift operation to access the turbine above sea-level. This can be a costly and lengthy procedure, resulting in prolonged periods of downtime. An effective condition monitoring system would therefore be of great benefit to this industry, allowing the health state of system components to be known, and allowing maintenance to be scheduled efficiently.

Condition monitoring has already been well established for the wind industry. However, despite similarities between tidal and wind power turbine design, the operating environment is vastly different. Water is over 800 times denser than air and, despite slower flow rates (around 3 m/s compared to around 15 m/s for offshore wind), tidal flow has a much higher kinetic energy compared to wind flow (Winter, 2011). This causes tidal turbines to operate with higher torque and thrust loading, inducing increased stress on the machine, particularly on the low speed stages of the drive train. Additionally, the marine environment provides other complications, such as corrosion and interaction with plant and animal life. Furthermore, there is limited historical data of failures from tidal turbines required to

Grant Galloway et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

implement condition monitoring techniques used as standard in the wind industry.

This paper focuses on using anomaly detection techniques for identifying developing faults within tidal turbines with limited historical data. Using the CRISP-DM data mining methodology (Wirth & Hipp, 2000), key relationships between sensor data parameters from an operational tidal turbine were identified, describing the normal response of the turbine over variable operating conditions. These trends were then defined using several modeling techniques, allowing for deviations from expected data patterns to be detected from live turbine data, alerting the operator to the possible onset of a fault. The implementation of an intelligent condition monitoring system is also discussed, to integrate a number of seperate models together through a decision system to assess the state of the turbine and its components.

1.1. HS1000 Turbine

The data examined within this paper was sourced from the Andritz Hydro Hammerfest HS1000 turbine (Figure 1). The HS1000 is an operational tidal turbine with a rated power of 1 MW, deployed off the coast of Orkney, Scotland, as part of the European Marine Energy Centre (EMEC).

Figure 1. The Andritz Hydro Hammerfest HS1000 tidal turbine

The turbine has an open-blade horizontal axis design, fixed to the seabed. Similar to a wind turbine, its drive train consists of a gearbox connected to an induction generator, translating tidal speeds of around 3.5 m/s to rotations exceeding 1000 RPM within the generator. The turbine has no yaw, with blades rotating in opposite directions in response to upstream and downstream tides. Pitch control of the blades is used to control the output power produced.

This paper will focus on data from the following sources:

- Tri-axial generator vibration velocity
- Gearbox vibration velocity
- Bearing vibration velocity
- Bearing displacement
- Bearing temperature
- Generator rotor speed
- Output power

1.2. Data Mining

Data mining is the analysis of large data sets for knowledge discovery. It involves the use of processing techniques, involving statistical, machine learning and visualization methods, to extract patterns and relationships hidden within data parameters (Olson & Delen, 2008). Data mining has been commonly used by banking and marketing firms, and also within the medical field applied to vast amounts of patient records for improved diagnosis and prediction (Maimon & Rokach, 2005).

Within this study, data mining was used to discover trends and relationships between parameters within initial datasets from the HS1000 tidal turbine. A modeling stage then defines the expected response of the turbine over its typical range of operating conditions. By comparing live turbine data to these models, anomaly detection is used to indicate a change in the response of the system, indicating the possible onset of a fault.

1.2.1. CRISP-DM

The CRISP-DM (Cross-Industry Standard Process for Data Mining) process model was utilized for this study. This model manages the data mining process as six key stages: business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Wirth & Hipp, 2000). These stages are shown in figure 2.

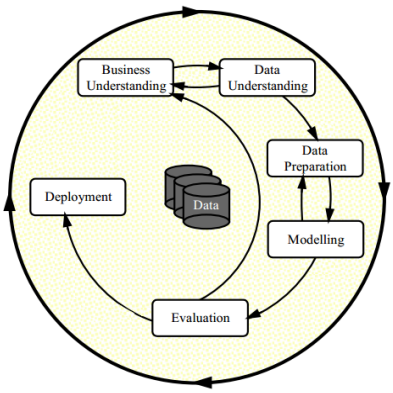


Figure 2. The CRISP-DM process model for data mining (Wirth & Hipp, 2000)

Each stage of the CRISP-DM process model was employed as follows:

- Business Understanding – Understand the operating environment of the turbine and how condition monitoring may be used to assess turbine health.
- Data Understanding – Use statistical analysis to identify key parameters, relationships, and trends to learn the

response of sensor data over standard operating conditions.

- Data Preparation – Organize sensor data before modeling, trending data and grouping by tidal cycle and operating state of the turbine.
- Modeling – Model key trends and relationships using curve fitting, Gaussian mixture modeling and kernel density estimation to define the response of data parameters over varying operating conditions.
- Evaluation – Evaluate the performance of each model, using past operational data to train and test models for anomaly detection.
- Deployment – Compare live data to models and identify deviations from expected behavior, integrating multiple models together through an intelligent condition monitoring system.

2. BUSINESS UNDERSTANDING

The business understanding phase of the CRISP-DM process model involved an appreciation of the operating environment and its effect on the expected response of the turbine. The role of condition monitoring within the field was also considered.

2.1. Condition Monitoring

The use of sensor data from turbine components (such as the gearbox, generator, bearings, blades, etc) can allow the onset of faults to be detected before they cause failure. This enables an efficient maintenance strategy to be employed, as maintenance can be scheduled to reflect to the known health of system components.

Examples of previous research on condition monitoring for tidal turbines includes:

- A review of condition monitoring and prognostic techniques applicable to tidal turbines (Wald, Khoshgoftaar, Beaujean & Sloan, 2010).
- Use of Failure Modes and Effects Analysis (FMEA) to detect faults and failures within tidal turbines (Prickett, Grosvenor, Byrne, Jones, Morris, O'Doherty & O'Doherty, 2011).
- Design of a dynamometer for simulating tidal turbine bearing faults, and application of wavelet based monitoring (Duhaney, Khoshgaftaar, Sloan, Alhalibi & Beaujean, 2011).
- Fatigue analysis of tidal turbine blades (Mahfuz & Akram, 2011).

However, since tidal turbines have limited deployment, there are few examples of condition monitoring systems implemented in practice reported in the literature.

2.2. Turbine Operation

The EMEC test site in Orkney experiences a semi-diurnal tide, with corresponding high and low tides each day. Upstream and downstream tidal flow is experienced by the HS1000 turbine in cycles between each high and low tide.

Figures 3 and 4 demonstrate the response of the turbine to a single tidal flow cycle, detailing generator rotor speed and output power. The rotation of the turbine is controlled through a combination of blade pitching and torque control through a frequency convertor. Generator rotor speed is held at approximately 800 RPM at low tidal flow rates, increasing to a value of over 1000 RPM as the flow rate increases. Output power varies more gradually with tidal flow rate, reaching a maximum of around 1 MW.

It is expected that these parameters will be most indicative of turbine operation, driving relationships with other data parameters as turbine components respond to changes in loading due to variation of tidal flow.

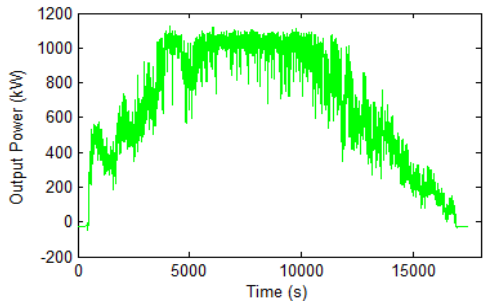


Figure 3. Trend of output power against time for a single tidal cycle.

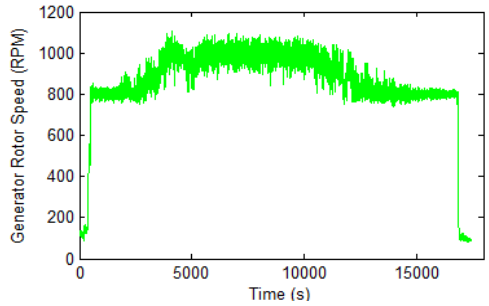


Figure 4. Trend of generator rotor speed against time for a single tidal cycle.

3. DATA EXPLORATION

Within this study, the data understanding stage of the CRISP-DM data mining process involved a statistic analysis of data parameters. Principal component analysis and correlation were used to reveal key relationships between parameters, indicative of normal operation of the HS1000 turbine over a range of operating conditions. This analysis also revealed differences in the response of the turbine to opposing tidal flow directions.

3.1. Principal Component Analysis

Principal component analysis (PCA) is a technique used to extract and remove linear correlations from a set of multivariate data (Pearson, 1901). This technique generates a set of principal components, which are the uncorrelated parameters underlying the observations within the data (Abdi & Williams, 2010).

Components are a list of coefficients, representing a weight for each input parameter, and an eigenvalue. Parameters with high weightings are the highest contributors to relationships within the data, and parameters with low weighting contribute the least. A component's eigenvalue is representative of the significance of a component to the data.

Results for this analysis returned components with high coefficient weightings for output power and generator rotation speed values, with high corresponding eigenvalues (in the range of $1x10^3$ to $1x10^5$). This confirmed these parameters were highly relevant within the data, driving relationships between other data parameters.

3.2. Correlation

Correlation describes the statistical relationship between two variables or data sets. This can be expressed via Pearson's correlation coefficient, which is a value describing the linear dependence of two parameters (Rodgers & Nicewander, 1988). This value ranges between +1 (an ideal increasing linear relationship) and -1 (an ideal decreasing linear relationship). Parameters with a correlation coefficient of zero have no association to each other.

Pearson's correlation coefficient was calculated for every pair of data parameters. High correlation was consistently seen in output power and generator rotor speed parameters, confirming these parameters are key to the response of other sensor data parameters (in particular gearbox and generator vibrations). Therefore, for the modeling stage of data mining, all other data parameters (including vibration, displacement and temperature readings from the gearbox, generator and bearings) were trended against output power and generator rotor speed. These relationships describe the response of turbine components over a range of varying operating conditions.

Comparison of these values also highlighted a change in system response between upstream and downstream tidal flows. This was expected as changes in tidal flow direction alter the direction of loads on the turbine. As a result, for the following stage of analysis, data was batched by tidal cycle and categorized by tidal flow direction. Separate models were then constructed to define the expected turbine response for both tidal flow directions.

3.3. Visual Analysis

Visual analysis confirmed meaningful relationships were generated by plotting data parameters against output power and generator rotor speed.

Trends against output power showed a spread of data across the full range of output power. This is expected, since the turbine generates at all tidal flow rates, and the output power is proportional to tidal flow. Figure 5 shows an example of gearbox vibration trended against output power for a single upstream tidal cycle.

Trends against generator rotor speed exhibited a less consistent spread of data, with points grouping in specific regions of the plot. This is because the generator rotor speed dictates the frequency of output power, which must be within defined limits to export power to the grid. Therefore, above the cut-in tidal flow rate, generator rotor speed increases immediately to approximately 800 RPM. Figure 6 shows an example of this trend, with generator vibration X-axis trended against generator rotor speed.

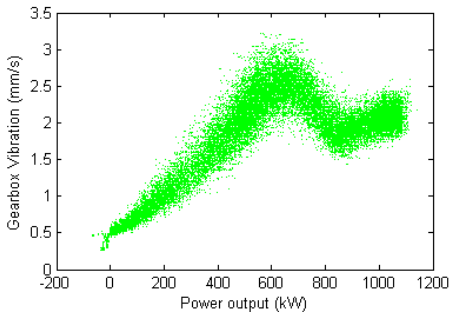


Figure 5. Trend of gearbox vibration against output power.

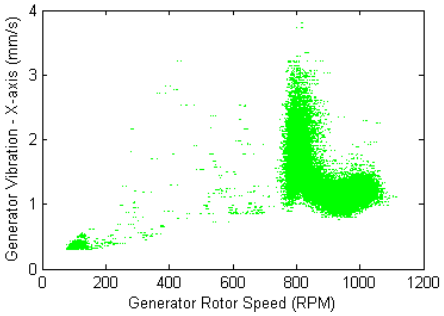


Figure 6. Trend of generator vibration X-axis against generator rotor speed.

4. DATA PREPARATION

The data preparation stage of the CRISP-DM model involved the organization of data before modeling, once key relationships had been identified.

Data was batched by tidal cycle, with upstream and downstream tidal flow data separated. Data parameters were then trended against output power and generator rotor speed.

Also at this stage, four key regions of data were defined, to further segment data before models were constructed. These regions were representative of the operating state of the turbine, and defined using change point analysis (Killick & Eckley, 2013) applied to the speed-power curve of the turbine.

4.1. Change Point Analysis

Change point analysis is a technique used to find a series of points within data parameters where changes in the data are most significant. Change points are determined by calculating a vector of the sum of differences between each data point and the mean of all data points. The maximum or minimum point on this vector will indicate the location of a change point (Killick & Eckley, 2013). This process can be repeated to find additional change points within each newly identified region.

Four regions of operation were visible from the speed-power curve (figure 7):

1. Start up and shut down region
2. Constant rotor speed region
3. Increasing rotor speed region
4. Turbine rotor speed and power limitation region

Figure 7 shows the result of change point analysis in defining these operating state regions. Separating these regions allowed the effects of the turbine's control scheme to be seen across other data parameters and was used to help partition data for use with anomaly detection techniques.

Figure 8 demonstrates how each operating region shapes the trend of gearbox vibration against output power. Changes in operating state can be clearly seen as maximum and minimum turning points in vibration level.

Figure 9 shows how groups of data are formed by the operating state of the turbine. Separating data points by operating regions allow these groups of data to be isolated and modeled separately.

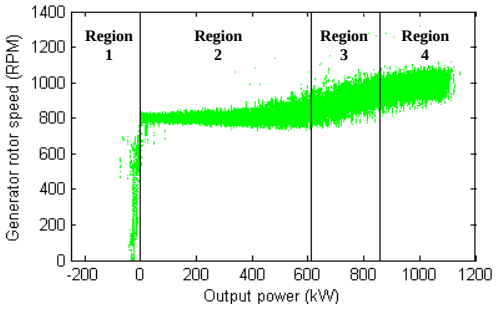


Figure 7. Turbine operating regions identified by change point analysis applied to speed-power curve.

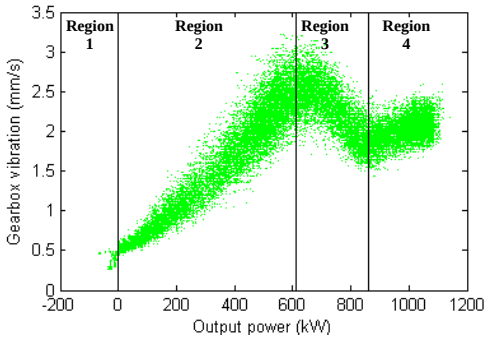


Figure 8. Turbine operating regions over gearbox vibration trended against output power.

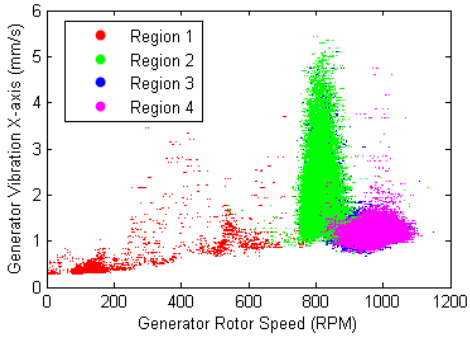


Figure 9. Generator vibration X-axis trended against generator rotor speed, separated by turbine operating region.

5. MODELING

With parameters trended against output power and generator rotor speed, a number of techniques were employed to best define the response of the system. Two types of relationships were observed between parameters: those where data was evenly spread throughout the trend, exhibiting patterns that could be modeled by an individual function; and those where data points tended to cluster within specific areas of a plot.

Curve fitting was used to define even spreads of data, fitting a function to the envelope of the trend or the entire trend itself. Within the data, this was applicable for vibration data trended against output power.

Gaussian mixture modeling and kernel density estimation techniques were used for defining relationships where data points clustered within specific areas. Clusters of data were separated by operating region (as in section 4.1), with areas defined probabilistically. This applied to parameters trended against generator rotor speed.

The output of this stage is a set of models that define the expected response of each turbine component. These models can then be used for anomaly detection, where live turbine data is compared to these models, and deviations represent the potential development of a fault within the system.

5.1. Curve Fitting

Curve fitting was applied to data parameters trended against output power, where relationships displayed an even spread of data across the trend. Initially, this technique was applied to the envelope of these trends, as maximum levels of vibration varied with output power. Anomalies would be detected in this case by data points exceeding maximum expected levels of vibration, lying about a curve fitted to the envelope.

Curve fitting was also applied to describe the trend between gearbox vibration and output power as a whole. This would enable additional metrics, such as variance, to be measured, with anomalies detected where data points exceeded a threshold of distance from the fitted curve.

Within this study, curve fitting was implemented in MATLAB using the 'Trust-Region-Reflective Least Squares' algorithm. This is an iterative method that tunes parameters $(\gamma_1, \gamma_2, \dots, \gamma_n)$ of the chosen function $f(\boldsymbol{x}, \boldsymbol{\gamma})$ to minimize the squared error between each data point (x_i, y_i) and the function itself, equation (1) (Hung, 2012).

$$\min_{\gamma} \sum_{i=1}^{m} \left(y_i - f(x_i, \boldsymbol{\gamma})\right)^2 \quad (1)$$

5.1.1. Envelope Fitting

Within the data from the HS1000 turbine, parameters trended against output power displayed varying levels of maximum vibration across their envelopes. Curves fitted to these envelopes will therefore describe a threshold of maximum expected vibration levels over the full range of turbine operation for each parameter, with anomalies detected above this threshold.

An envelope was determined by sampling maximum values of output power across a trend. A curve was then fitted to this envelope, describing the expected boundary of a data parameter. Each stage of this process is outlined in figure 10.

Functions were chosen to model each parameter trended against output power that minimized the root mean squared error (RMSE) between the function and the envelope. Table 1 summarizes the RMSE values for Gaussian and Polynomial functions of increasing orders fitted to the envelope of generator vibration Z-axis trended against output power. Gaussian functions of varying order were found to best fit the envelopes of all parameters trended against output power, with the order number representative of the number of peaks across the envelope.

Table 1. Summary of RMSE values for curve fitting applied to envelope of generator vibration Z-axis trended against output power.

Function	RMSE
6th order Polynomial	0.732
7th order Polynomial	0.727
8th order Polynomial	0.719
9th order Polynomial	0.722
3rd order Gaussian	0.761
4th order Gaussian	**0.709**
5th order Gaussian	0.782

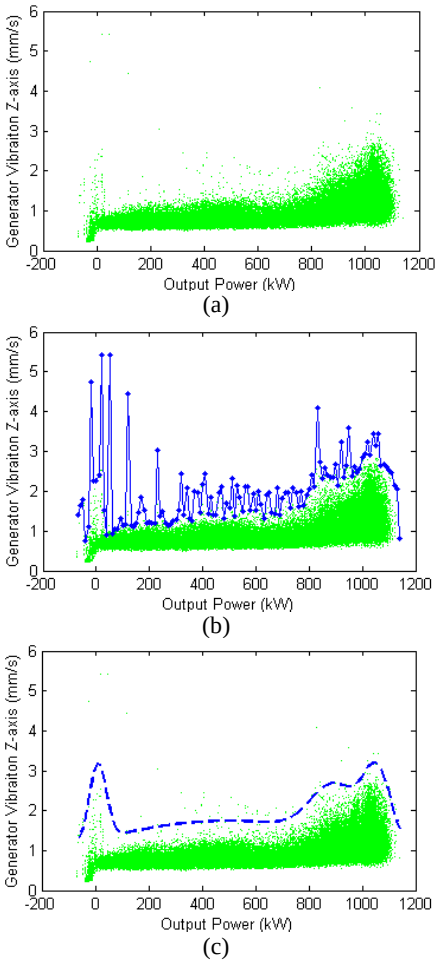


Figure 10. Curve fitting applied to the envelope of a generator vibration Z-axis trended against output power. (a) Data parameter trended against output power. (b) Sampled envelope across trend. (c) 4th order Gaussian function fitted to envelope.

5.1.2. Gearbox Vibration Curve Fitting

Gearbox vibration parameters trended against output power displayed all data points varying across the relationship (figure 5). Fitting a curve to describe the relationship as a whole would allow for additional measures to be determined, such as variance, to reveal additional information about the response of the system. Anomalies would be detected in this case by data points exceeding a certain distance from the fitted function.

A number of different functions were fitted to this relationship, observed by three seperate vibration sensors. Table 2 summarizes the results, including Gaussian and polynomial functions, as well as a piecewise linear fit within each defined operational region (i.e. four sequential linear fits). The accuracy of each function was compared using the RMSE value between the function and all data points used to generate the model.

Table 2. Summary of RMSE values for curve fitting applied to the trend of gearbox vibration against output power

Function	RMSE		
	Gearbox vibration sensor 1	Gearbox vibration sensor 2	Gearbox vibration sensor 1
Linear fit between operating regions	0.204	0.209	0.243
6th order Polynomial	0.221	0.218	0.238
3rd order Gaussian	**0.203**	**0.205**	**0.237**

The Gaussian function was found to best describe this relationship returning the lowest RMSE. This is shown in figure 11.

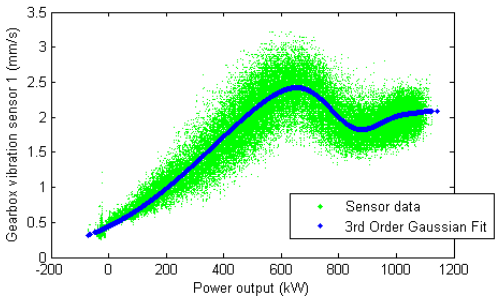


Figure 11. 3rd order Gaussian function fitted to gearbox vibration sensor 1 trended against output power.

5.2. Gaussian Mixture Modeling

Since the generator rotor speed does not behave as a continuous variable (unlike output power), curve fitting approaches are less appropriate. Two techniques were employed to define the operational groups of data against generator rotor speed: Gaussian mixture modeling and kernel density estimation. Each technique defined regions of data by probability. Deviations from expected response of the turbine can be identified as live turbine data occurring with low values of probability when compared to these models.

Gaussian mixture modeling is a method used to fit a combination of n-dimensional Gaussian distributions, each with a given weighting, to an n-dimensional data set (Dempster, Laird & Rubin, 1977). This was performed in MATLAB through the Expectation Maximisation algorithm (Bilmes, 1998). This method involves making an initial 'guess' (randomly generated within a given range) of Gaussian parameters, and calculating the probability of the data points within this model. The model parameters are then updated iteratively to maximize the likelihood of each data point. This process is stopped once a threshold of convergence is reached.

Figure 12 details the result of Gaussian mixture modeling within a contour plot for the Z-axis component from the generator vibration sensor trended against generator rotor speed, separated into the four operating regions. Within this plot, outwardly lines represent areas of decreasing probability. These plots revealed this method works well for regions 2, 3 and 4, where contour lines fit tightly around clear groups of data. However, this technique is not as effective for region 1, where data points are spread more sparsely throughout the plot.

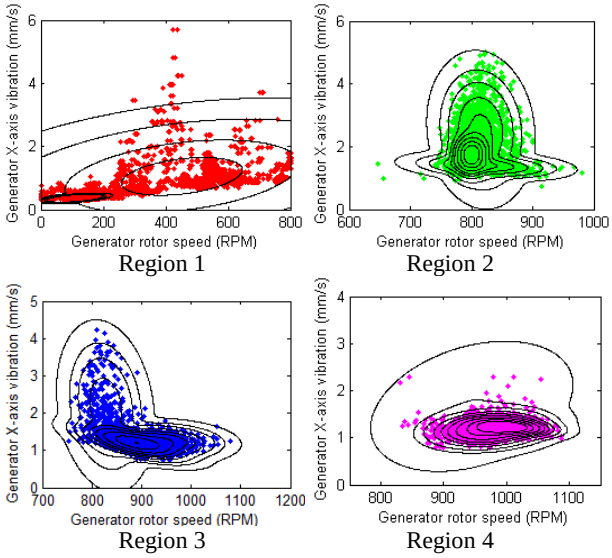


Figure 12. Contour plot of Gaussian mixture modeling applied to the trend between generator vibration X-axis and generator rotor speed.

5.3. Kernel Density Estimation

Gaussian kernel density estimation is a technique similar to Gaussian mixture modeling, used to the same effect within

this study to define regions of probability between parameters. However, this technique differs as it aims to approximate the true probability density function (PDF) of the data.

The true distribution is estimated by computing the sum of small individual PDFs at each observed data point (Zucchi, 2003). In this case, the Gaussian distribution was used as the individual (kernel) PDF. This method will generate a more accurate model, however it is a lot more computationally intensive. This was implemented in MATLAB by adapting a method by Cao (2013).

Figure 13 shows a contour plot describing Gaussian kernel density estimation applied to the Z-axis component from the generator vibration sensor trended against generator rotor speed, separated into the four operating regions. In comparison to Gaussian mixture modeling (figure 12), this technique provides a much closer fit to the data, particularly within region 1.

Although this model was more accurate, it produces a less general model, treating individual data points lying outside the main group of data as separate regions of data. This model can be improved by training with as many datasets as possible. It is expected that additional data points will fill in some of the spaces between separately defined regions. Alternatively, smaller groups of data could be removed in pre-processing, and the resultant model could be smoothed over a larger area.

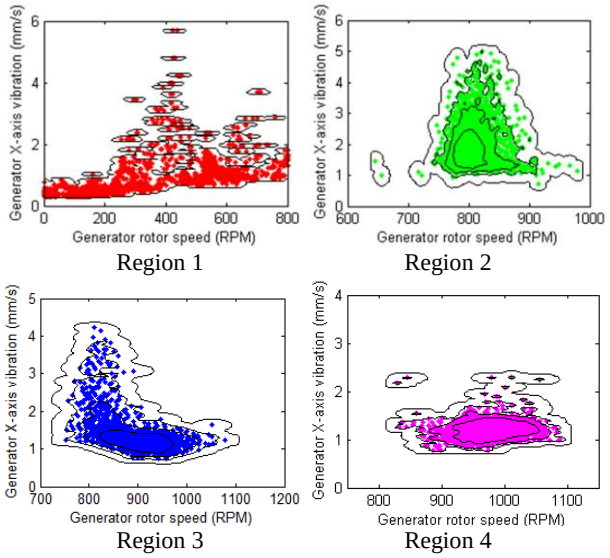


Figure 13. Contour plot of Gaussian kernel density estimation applied to the trend between generator vibration X-axis and generator rotor speed.

6. EVALUATION

Using the techniques described above, models were constructed using training data from October 2013, and tested using data from subsequent December, January and February. Appropriate metrics were then extracted to detect anomalies and measure the severity of deviations from training data. Although no fault data was available, results showed the effectiveness of each technique in defining system behavior and observing changes over time.

6.1. Envelope Fitting

Envelope fitted models, describing parameters trended against output power (as in section 5.1.1.) were tested, where anomalies were detected as data points exceeding the Gaussian function used to describe the envelope of training data.

Some anomalies were detected as crossing the boundary, shown in figure 14. Using this technique a number of metrics can be extracted, including number of anomalies, percentage of anomalies and average distance from the boundary, to indicate the severity of a deviation from normal behavior.

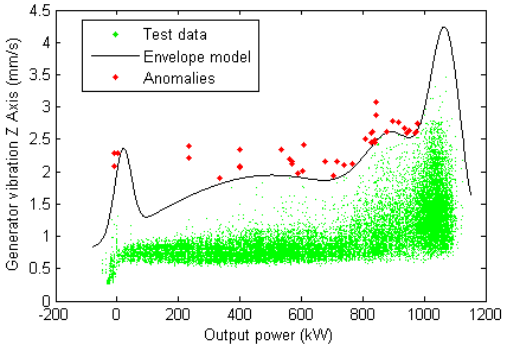


Figure 14. Envelope fitting anomaly detection applied to December 2013 test dataset.

Table 3 summarizes the results of testing this technique, with each metric measured for each test dataset. No significant deviations were detected in the test datasets, with the total number of anomalies being minimal and average distances not being significantly large. This correctly suggests normal behavior.

Table 3. Envelope fitting test results

Testing dataset	Number of anomalies	Percentage of anomalies	Average distance (mm/s)
December 2013	156	0.1718 %	0.226
January 2014	36	0.0361 %	0.388
February 2014	69	0.0418 %	0.355

6.2. Curve Fitting

The curve fitting modeling technique was tested on the relationship between gearbox vibration and output power, using a 3rd order Gaussian curve (as in section 5.1.2.).

Figure 15 shows the December 2013 testing data compared against a trained model constructed from October 2013 data. In contrast to envelope model fitting, no set boundary is used to indicate anomalous data points. Instead, metrics such as maximum error and RMSE can be used to measure the severity of any deviation from normal system response.

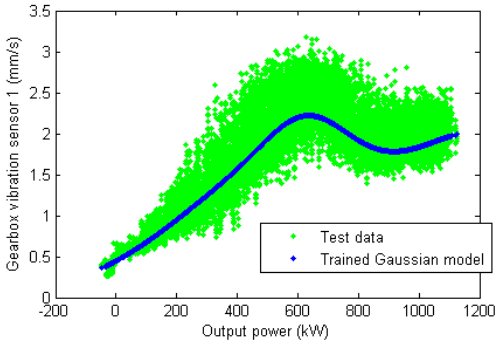


Figure 15. Curve fitting anomaly detection applied to December 2013 test data.

Table 4 summarizes testing results using these metrics. An increase in RMSE is seen in both December and February where more data points are lying above the Gaussian function, indicating an overall increase in vibration across the full operating range. This was attributed to seasonal changes in tidal flow affecting the test data, and not component wear or damage.

Table 4. Curve fitting test results

Training dataset	**Max Error (mm/s)**	**RMSE (mm/s)**
October 2013	1.45	0.203
Testing dataset	**Max Error (mm/s)**	**RMSE (mm/s)**
December 2013	1.26	0.251
January 2014	0.93	0.202
February 2014	1.04	0.235

6.3. Gaussian Mixture Modeling

Gaussian mixture modeling was tested on clusters of generator vibration data trended against generator rotor speed, separated by operational regions, as described in sections 4.1. and 5.2. Results detailed in this section were recorded from the generator X-axis vibration parameter.

Anomalies were considered to be data points lying outside the 95% confidence interval. The percentage of anomalies lying outside the 95% confidence interval (CI) was used as a metric. A value exceeding 5% was considered to indicate that the model was not a good fit to the test data and a change in system response may have occurred.

Table 5 and figure 16 show the results of testing. A number of clusters were identified to have a significant number of anomalies, with percentages exceeding 5%. These results indicate a deviation in system response over time, however, the variations were due to seasonal changes in tidal flow. The significant number of anomalies is therefore not representative of the relatively small variation in data, and it was concluded that Gaussian mixture modeling provided a poor representation of training data distributions.

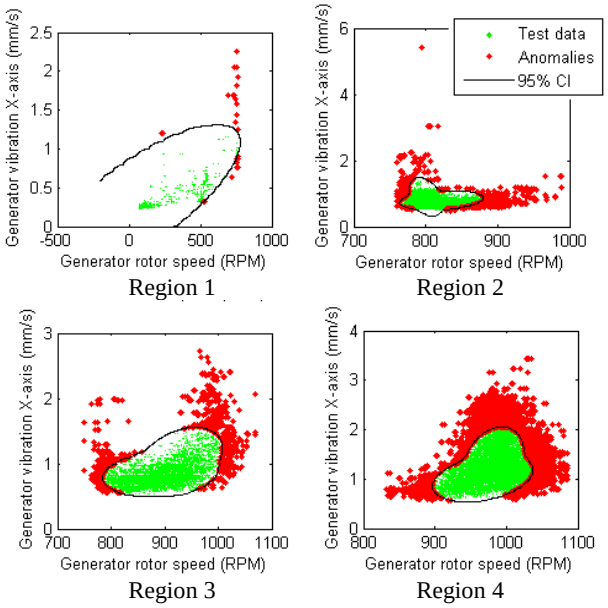


Figure 16. Gaussian mixture modeling anomaly detection applied to December 2013 test data.

Table 5. Gaussian mixture modeling test results

Testing dataset	**Region**	**No. of Anomalies outside 95% CI**	**Percentage of anomalies**
December 2013	1	59	1.006
	2	1532	4.454
	3	1425	9.799
	4	7180	19.928
January 2014	1	88	0.821
	2	347	1.998
	3	194	1.552
	4	10350	19.938
February 2014	1	2469	5.776
	2	6031	5.519
	3	1986	15.055
	4	19	52.777

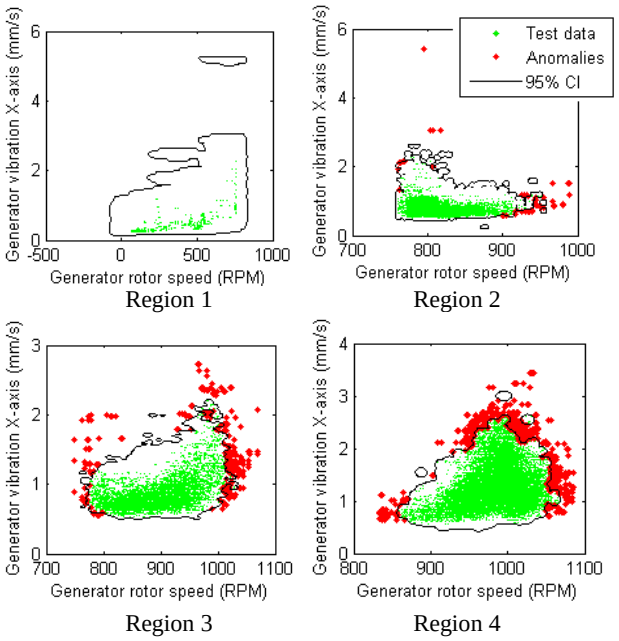


Figure 17. Kernel density estimation anomaly detection applied to December 2013 test data.

Table 6. Kernel density estimation test results

Testing dataset	Region	No. of Anomalies outside 95% CI	Percentage of anomalies
December 2013	1	0	0
	2	108	0.314
	3	499	3.441
	4	1085	3.011
January 2014	1	4	0.037
	2	6	0.025
	3	35	0.280
	4	1532	2.871
February 2014	1	0	0
	2	99	0.090
	3	777	5.894
	4	10	2.778

6.4. Kernel Density Estimation

Kernel density estimation was tested on clusters of data separated by operating regions as in 6.3. As with Gaussian mixture modeling, anomalies were detected as data points lying outside the 95% confidence interval (CI). Here, the confidence interval was calculated using bootstrap sampling, as by Chen, Goulding, Sandoz and Wynne (1998).

Table 6 and figure 17 show the results of testing. In contrast to results achieved through Gaussian mixture modeling, the number of detected anomalies is significantly less, and under 5% in the majority of cases. This suggests kernel density estimation provides a more accurate representation of the distribution of data points within each cluster and is therefore a more suitable technique for this application.

6.5. Summary of Results

Results were obtained to test the effectiveness of a number of modeling techniques, used to define the expected response of a tidal turbine under normal operating conditions.

Envelope and curve fitting techniques were observed to provide a good representation of expected turbine response, capable of detecting small seasonal deviations in data over time. Gaussian mixture modeling was seen to be less effective, detecting a large number of anomalies where little deviation occurred. Kernel density estimation was favored over this technique.

Each anomaly detection technique provides a seperate group of metrics to describe anomalous behavior, indicating the severity of deviations. Future work will involve the analysis of further metrics to support testing on additional data as it becomes available.

No significant changes in system response were observed within test data, with only small variations seen due to seasonal changes in tidal flow. It is therefore recommended that models are examined at regular monthly intervals with deviations matched against seasonal trends. Models can then be updated accordingly.

7. DEPLOYMENT

Since there is no single model which covers all parameter relationships, the deployment stage involves using each model of expected turbine response in parallel to perform anomaly detection of live turbine data. Each individual model can be integrated as part of an intelligent system, with seperate models implemented to process data from the turbine and output whether or not the data has deviated from expected trends. A decision system linked to these modules can then use these results to make assessments of turbine health.

Although anomaly detection is useful as an initial stage of condition monitoring, it is only suitable for indicating if a deviation from the defined normal behavior has occurred. Specific failure modes cannot be identified through this method. Further stages of condition monitoring include diagnosis and prognostics.

Diagnosis involves analysis of turbine component failure modes, and understanding how these will be represented within data parameters. Prognostics involve assessing the current state of the system and estimating the remaining useful life of individual components, or the system as a whole. Various algorithms can be used for these purposes, including those used within machine learning and artificial intelligence, such as neural networks or Bayesian classifiers. Future work will explore these algorithms in relation to the system, utilizing failure data as it becomes available. Diagnostics and prognostics can be implemented as additional modules in the intelligent system.

8. CONCLUSION

This paper outlined the use of data mining through the CRISP-DM process model to explore data from the HS1000 tidal turbine and define its expected operational behavior. The use of principal component analysis and correlation revealed key relationships within the data, relating parameters to output power and generator rotor speed.

Envelope and curve fitting techniques were found to provide accurate models of the response of system components to changes in output power. Kernel density estimation was also found to be an effective technique when used to model clusters of generator vibration data formed when trended against rotor speed. Gaussian mixture modeling was found to be less effective in this application.

Models were trained using past operational turbine sensor data, with anomaly detection performed using data from subsequent months. Small deviations in system response were detected, due to seasonal changes in tidal flow.

Future work will involve the analysis of further metrics to describe the severity of anomalous responses, using additional data as it becomes available. Once techniques are established, an intelligent condition monitoring system will be designed to integrate seperate modules together and assess the state of the turbine and its components. With further research, additional modules can be added to the intelligent system, to perform diagnosis and prognosis as failure data becomes available.

9. REFERENCES

Abdi, H. & Williams, L. (2010). Principle Component Analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), pp. 433-459

Aly, H. H. H., El-Hawary, M. E. (2011). State of the Art for Tidal Currents Electric Energy Resources. *24th Canadian Conference on Electrical and Computer Engineering (CCECE)* (1119-1124), May 8-11, Niagra Falls, ON. doi:10.1109/CCECE.2011.6030636

Bilmes, J. (1998). A Gentle Tutorial of the EM algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *Technical Report ICSI-TR-97-021*, University of Berkeley

Cao. Y. (2013). Bivariant Kernel Density Estimation (V2.1). http://www.mathworks.co.uk/matlabcentral/fileexchange/19280-bivariant-kernel-density-estimation-v2-1/content/html/gkde2test.html

Chen, H., Ait-Ahmed, N., Zaim, E. & Machmoum, M. (2012). Marine Tidal Current Systems: state of the art. *2012 IEEE International Symposium on Industrial Electronics* (1431-1437), May 28-31, Hangzhou, China. doi:10.1109/ISIE.2012.6237301

Chen, Q., Goulding, P., Sandoz, D. & Wynne R. (1998). The Application of Kernel Density Estimates to Condition Monitoring for Process Industries. *Proceedings of the American Control Conference* (pp 3312-3316). June 1998, Philadelphia, PA. doi: 10.1109/ACC.1998.703187

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1997). Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), pp. 1-38. doi: 10.1.1.133.4884

Duhaney, J., Khoshgoftaar, T. M., Sloan, J. C., Alhalabi, B. & Beaujean, B. (2011). A Dynamometer for an Ocean Turbine Prototype – Reliability through Automated Monitoring. *2011 IEEE 13th International Symposium on High-Assurance Systems in Engineering* (pp 244-251). November 10-12, Boca Raton, FL. doi: 10.1109/HASE.2011.61

European Union Committee (2008). The EU's Target for Renewable Energy: 20% by 2020, *27th Report of Session 2007-08*

Hung, J. (2012). *Energy Optimization of a Diatomic System*. University of Washington, Seattle, WA. http://www.math.washington.edu/~morrow/papers/jane-thesis.pdf

Killick, R. & Eckley, I. A. (2013). *Changepoint: An R Package for Changepoint Analysis*. Lancaster University, UK.

King, J. & Tryfonas, T. (2009). Tidal Stream Power Technology – State of the Art. *OCEANS 2009 – EUROPE* (1-8). May 11-14, Bremen. doi:10.1109/OCEANSE.2009.5278329

Mahfuz, H. & Akram, M. W. (2011). Life Prediction of Composite Turbine Blades under Random Ocean Current and Velocity Shear. *OCEANS 2011 IEEE – SPAIN* (1-7). June 6-9, Santander. doi:10.1109/Oceans-Spain.2011.6003526

Maimon, O., Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*. New York, NY: Springer

Olson, D. L., Delen, D. (2008). *Advanced Data Mining Techniques*. Heidelberg: Springer

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(11), pp. 559-572

Prickett, P., Grosvenor, R., Byrne, C., Jones, A. M., Morris, C., O'Doherty, D. & O'Doherty, T. (2011). Consideration of the Condition Based Maintenance of Marine Tidal Turbines. *9th European Wave and Tidal Energy Conference*. September 5-9, Southampton UK.

Rodgers, J. L. & Nicewander, W. A. (1988). Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1), pp. 59-66

Wald, R., Khoshgoftaar, T. M., Beaujean, B. & Sloan, J. C. (2010). A Review of Prognostics and Health Monitoring Techniques for Autonomous Ocean Systems. *16th ISSAT International Conference Reliability and Quality in Design*. August 5-7, Washington, D.C.

Winter, A. I. (2011). Differences in Fundamental Design Drivers for Wind and Tidal Turbines. *OCEANS, 2011 IEEE – SPAIN* (1-10), June 6-9, Santander. doi: 10.1109/Oceans-Spain.2011.6003647

Wirth, R. & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining* (29-39), Manchester, UK.

Zucchi, W. (2003). *Applied Smoothing Techniques, Part 1: Kernel Density Estimation*. Philadelphia, Pa: Temple University

BIOGRAPHIES

Grant S. Galloway is a PhD student within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his M.Eng in Electronic and Electrical Engineering from the University of Strathclyde in 2013. His PhD focuses on condition monitoring and prognostics for tidal turbines, in collaboration with Andritz Hydro Hammerfest, a leading tidal turbine manufacturer.

Victoria M. Catterson is a Lecturer within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. She received her B.Eng. (Hons) and Ph.D. degrees from the University of Strathclyde in 2003 and 2007 respectively. Her research interests include condition monitoring, diagnostics, and prognostics for power engineering applications.

Craig Love is a Turbine System Engineer with Andritz Hydro Hammerfest, Glasgow, UK.

Andrew Robb is a Mechanical Engineer with Andritz Hydro Hammerfest, Glasgow, UK.

Detection of Wind Turbine Power Performance Abnormalities Using Eigenvalue Analysis

Georgios Alexandros Skrimpas[1], Christian Walsted Sweeney[2], Kun S. Marhadi[3], Bogi Bech Jensen[4], Nenad Mijatovic[5], and Joachim Holbøll[6]

[1,2,3] *Brüel and Kjær Vibro, Nærum, 2850, Denmark*
alexandros.skrimpas@bkvibro.com
christian.sweeney@bkvibro.com
kun.marhadi@bkvibro.com

[4] *University of the Faroe Islands, Tórshavn, 100, Faroe Islands*
bogibj@setur.fo

[5,6] *Technical University of Denmark, Lyngby, 2800, Denmark*
nm@elektro.dtu.dk
jh@elektro.dtu.dk

ABSTRACT

Condition monitoring of wind turbines is a field of continuous research and development as new turbine configurations enter into the market and new failure modes appear. Systems utilising well established techniques from the energy and industry sector, such as vibration analysis, are commercially available and functioning successfully in fixed speed and variable speed turbines. Power performance analysis is a method specifically applicable to wind turbines for the detection of power generation changes due to external factors, such as icing, internal factors, such as controller malfunction, or deliberate actions, such as power de-rating. In this paper, power performance analysis is performed by sliding a time-power window and calculating the two eigenvalues corresponding to the two dimensional wind speed - power generation distribution. The power is classified into five bins in order to achieve better resolution and thus identify the most probable root cause of the power deviation. An important aspect of the proposed technique is its independence of the power curve provided by the turbine manufacturer. It is shown that by detecting any changes of the two eigenvalues trends in the five power bins, power generation anomalies are consistently identified.

Georgios Alexandros Skrimpas et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Nowadays, condition monitoring of wind turbines is directly connected to the predictive maintenance strategy employed by numerous operators in order to increase the availability, minimize the maintenance expenses, reduce the downtime and therefore the cost of energy (CoE) (Butler, Ringwood, & O'Connor, 2013). As many countries in Europe and worldwide have set high goals for the renewable energy penetration on their systems, CoE constitutes an important parameter for the competitiveness of wind power compared to the conventional energy sources (Lu, Li, Wu, & Yang, 2009).

Techniques such as vibration, temperature and oil analysis have been extensively applied for the mitigation of the unexpected operation and maintenance expenses over the past years focusing mainly on the drive train components. Continuous data trending is an essential part of condition monitoring in order to identify the commence of a faulty state and its progression in time. A typical example is the trending of speed related narrowband filtes, such as running speed harmonics and tooth mesh frequencies, and not speed related broadband measurements in vibration analysis (Marhadi & Hilmisson, 2013).

As the power rating of modern turbines is continuously increasing reaching 8MW in prototype installations, it is a requirement that their condition monitoring is performed holistically combining various techniques. Power performance analysis can be used as an assisting tool along with the established methods, such as vibration analysis. Its utilization as power generation abnormality detector and general indica-

tor of the overall health of the turbine is based on analysing standard collected supervisory control and data acquisition (SCADA) system information and extracting useful features (Uluyol, Parthasarathy, Foslien, & Kim, 2011).

The theoretical input power obtained from wind can be expressed by the following equation:

$$P = 0.5\rho A C_p(\lambda, \beta) u^3 \tag{1}$$

where P is the power captured by the wind turbine rotor, ρ is the air density, A is the swept rotor area, C_p is the power coefficient, β is the blade-pitch angle, λ is the tip-speed ratio and u is the wind speed (Lydia, Selvakumar, Kumar, & Kumar, 2013). Furthermore, the air density ρ is equal to:

$$\rho = \frac{p}{RT} \tag{2}$$

where p is the absolute air pressure and R is the specific gas constant; these two parameters are functions of altitude and humidity (Schlechtingen, Santos, & Achiche, 2013). Finally, the air density ρ is also influenced by the ambient temperature T.

The above equations suggest that the input wind power depends on the weather conditions (seasonality) and the site of erection. Other factors, such as terrain, park topology, and wake effects contribute on the unique power production profile of every turbine (Mchali, Barthelmie, Frandsen, Jensen, & Rthor, 2006). Therefore, utilization of the nominal power curve applicable to each turbine type enhances a number of challenges which may complicate the identification of abnormalities.

In addition to the above, the wind turbine power production can be affected by external factors, such as icing and dirt on blades; internal factors, such as pitch system defect or control system malfunction; or by deliberate actions, such as power de-rating or application of specific operation modes (Park, Lee, Oh, & Lee, 2014). The aforementioned conditions yield power generation deviations which can be observed in different power production states.

In this paper, the application of eigenvalue analysis for monitoring of power performance deviations due to external factors and deliberate actions is presented and analysed. There are two special points on the proposed performance assessment method. Firstly, the power curve is divided in discrete power classes deviating from the conventional approach of having wind bins (Park et al., 2014). The power classification is followed in order to obtain finer resolution so as to discriminate between different performance deterioration factors. Furthermore, eigenvalue analysis is an unsupervised method meaning that the objective is to calculate a number of features from the distribution under consideration rather than explicitly defining relations between sets of variables, e.g. condition distributions in the form $p(output|input)$. Hence, prior knowledge of the power curve suggested by the wind turbine manufacturer or employment of power curve learning are not required.

The paper structure is as follows. Section 2 provides a short description to the mathematical background of eigenvectors and eigenvalues. In section 3, the method description is presented based on the analysis of a turbine subjected to ice build-up. The trending behaviour of the calculated eigenvalues is illustrated in section 4 for the cases of icing, power de-rating and operation under noise reduction mode. Finally, sections 5 and 6 present the discussion and conclusions respectively.

2. EIGENVECTORS AND EIGENVALUES BACKGROUND

The statistical characteristics of a given data set can be represented by the covariance matrix, its eigenvalues, and the corresponding eigenvectors. The following analysis is classified as an unsupervised learning method which can be used to discover correlation among patterns as well as intrinsic directions where the data patterns change most (with maximum variance).

$\mathbf{R_{xx}}$ is defined as the covariance matrix of the power curve data set, with dimension $N = 2$. The two orthonormal eigenvectors $\mathbf{e_1}$ and $\mathbf{e_2}$, corresponding to the eigenvalues λ_1 and λ_2 of the data covariance matrix $\mathbf{R_{xx}}$ are called eigenvectors.

$$\mathbf{R_{xx}} \cdot \mathbf{e}_i = \lambda_i \cdot \mathbf{e}_i \quad , \quad i = 1, 2 \tag{3}$$

These eigenvectors show orthogonal directions in the pattern space where data change is maximum (maximum variance) (Cios, Pedrycz, Swiniarski, & Kurgan, 2007). The latter feature is used to explore any abnormal deviations of the power curve which could potentially correspond to power production anomalies.

Providing a two dimensional data set (wind speed and power production), the number of eigenvectors is two. However, if more data related to the wind turbine operation are taken into consideration, such as the blade pitch angle, the rotor running speed, the ambient temperature and the nacelle direction, then principal component analysis could be employed to extract only the most informative factors. This reduction of dimensionality is usually applied on classification problems for data compression (Bishop, 2006).

3. WIND TURBINE POWER PERFORMANCE MONITORING VIA EIGENVALUES VARIATIONS

Figure 1 depicts the power production and wind speed s function of time for Turbine#14 for a period of approximately two years along with the derived power curve. The power produc-

tion - wind speed data are sampled every one hour. The negative power values correspond to periods where the turbine is set to local mode due to performed maintenance activities or inspection of potential faulty components.

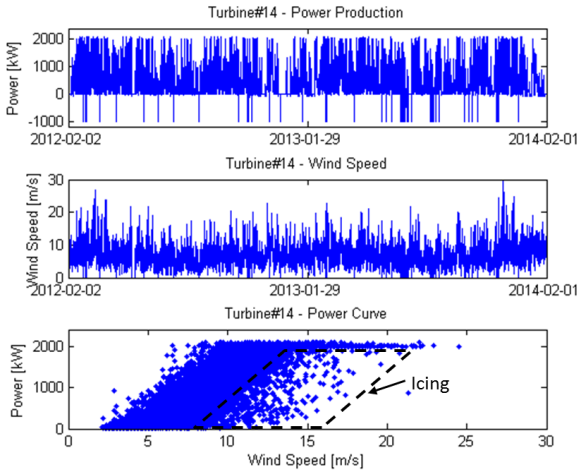


Figure 1. Turbine#14 - Power Production, Wind Speed and Power Curve - Case: Ice build-up on blades.

In order to detect any power performance changes, a sliding time window is used. The time window length selection is a compromise between computational cost and capability of extracting useful information. A reasonable choice is between one to three weeks, as a too long window would result in smoothing phenomena and a too short window would generate noisy results. The sliding time window can be overlapping for finer time resolution. The overlapping selection is also a function of the computational cost and desired time step. The analysis in the following sections is based on time window of two weeks and time step of one hour.

In order to proceed to the recognition of any patterns efficiently, the sliding time window is further divided into five power bins (classes). The classification into five bins follows Brüel and Kjær Vibro's vibration based condition monitoring scheme (Andersson, Gutt, & Hastings, 2007). The five classes are evenly distributed in general terms, but they might alter for different turbine models. The power classification is implemented so as to distinguish between various factors influencing the power production.

Figure 2 presents the power curve points of Turbine#14 under normal and abnormal power production for two weeks in late September 2013 and mid January 2014, along with the nominal power curve provided by the turbine manufacturer (black dashed line). The abnormal operation is due to ice build-up on the turbine blades, which was verified by the park operator. For better illustration, figure 3 presents the contour plot of the two dimensional histogram corresponding to the data shown in figure 2. The red lines correspond to high probability density function (pdf) values whereas the blue lines indicate low pdf values.

The data distribution of the right subplot in low to mid power production is significantly shifted to the right compared to the left subplot as well as compared to the power curve provided by the manufacturer. However, it should be noted that the ideal power curve should not be fully trusted as it is a function of the air density and consequently of the ambient temperature, which is not available for this turbine. Furthermore, it should be emphasized that the performance of a wind turbine is also influenced by site related factors and thus any discrepancies are not necessarily indicators of abnormal behaviour.

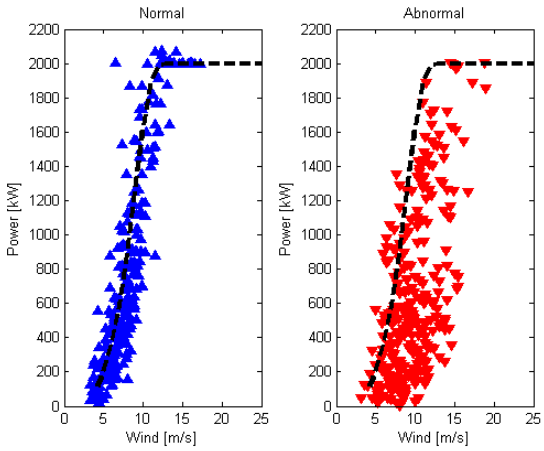


Figure 2. Turbine#14 - Power curve points under normal and abnormal (icing) power production.

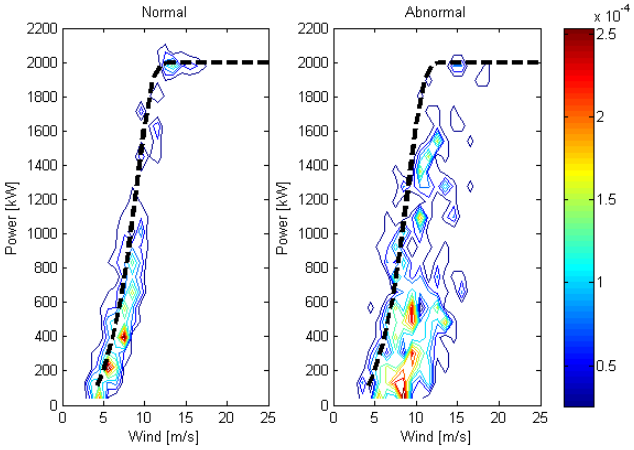


Figure 3. Turbine#14 - Contour plot of two dimensional histogram under normal and abnormal (icing) power production.

Following the power classification approach, figure 4 presents the contour plot of the two dimensional histogram in low production, i.e. from 0% to 30% of the nominal power output, for

both normal and abnormal operation. It can be noticed that two orthonormal vectors are included for the two cases under investigation. The two vectors are further described by two quantities, direction and magnitude. The direction is defined by the eigenvector and the magnitude by the corresponding eigenvalue. The eigenvalues represent the variances of the data set in directions specified by the eigenvectors. Given that the direction does not vary significantly, the eigenvalues provide essential information about the scatter of the distribution and consequently the power performance of the wind turbine. Hence, figure 4 suggests that the distribution presented in the right subplot is drawn from a wind speed - power production data set where the performance of the turbine is influenced by an external factor. Bearing in mind that the right set corresponds to two weeks in January 2014 and that the turbine is installed in cold climate location, it can be concluded that icing is the most likely root cause of the detected power curve deviation.

The naming convention wind and power variation is adopted for the two eigenvalues. The virtual unit for wind variation is in m/s and for power variation is in kW.

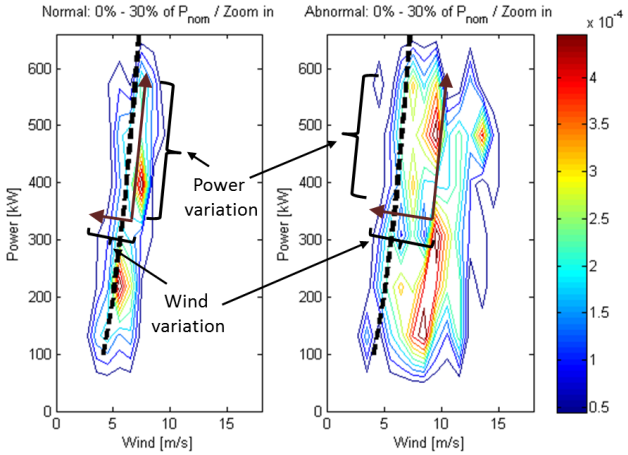


Figure 4. Turbine#14 - Zoom in low power production contour plot of two dimensional histogram under normal and abnormal (icing) power production.

4. DETECTION OF WIND TURBINE POWER PERFORMANCE ABNORMALITIES

Figures 5 and 6 show the trending behaviour of the square root of the two eigenvalues for two power classes, 0%-30% and 30%-50% of the rated power output. The sliding window length is two weeks and the time step is set to one hour.

It can be observed that the wind variation shows increased trends in both power bins in winter seasons. The increase in the trends shows that the scatter of the two-week sets is wider, indicating potential performance deterioration. Especially in winter 2012-2013, one can notice several hills and

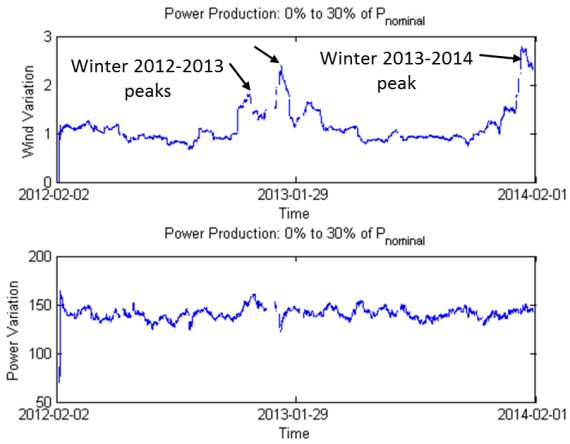


Figure 5. Turbine#14 - Icing - Trending behaviour of eigenvalues in low power production.

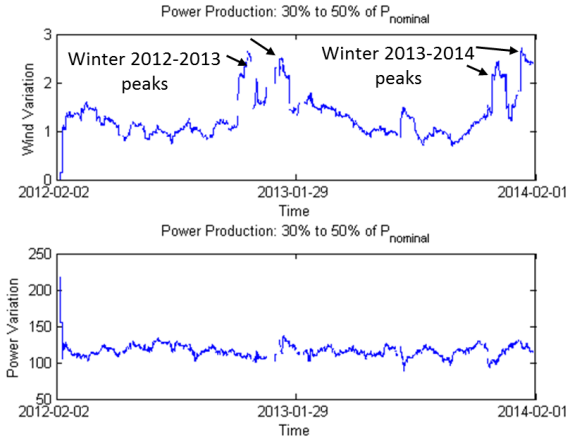


Figure 6. Turbine#14 - Icing - Trending behaviour of eigenvalues in low to mid power production.

valleys. The cause was ice formation on the turbine blades in December 2012, which was successfully removed by the turbine operator. However, the turbine was subjected to icing again a few days later resulting in emergency stop. The same phenomenon was repeated in winter 2013-2014, where again the wind variation behaviour presents clear increasing trends.

The above example focuses on icing detection, which can be classified as a condition which needs to be addressed by the turbine operator. However, many reasons, such as power derating or enabling of certain operation modes, can change power production from expected. If these actions are not communicated properly between the involved parties (park supervisor and technicians, performance centre, condition monitoring supplier) or the information flow has a delay of several days, unnecessary processes may initiate from either party.

Figure 7 presents the power production, wind speed time series and power curve of Turbine#09. The power output has

been de-rated two times over the past two years due to grid issues. The power curve subplot validates the above as a cluster of points is centred at 1.5MW for wind speed above $12m/s$.

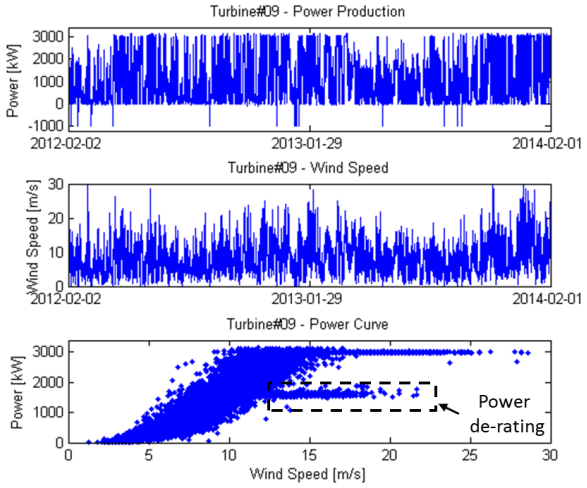


Figure 7. Turbine#09 - Power Production, Wind Speed and Power Curve - Case: Power output de-rating.

By inspecting the power production over time, one could identify that the generated power was restricted to approximately 50% in the beginning of 2013 until middle of the year. Although the wind speed could be advised to verify the above, the procedure is time consuming as data of at least a few days shall be available for confirmation. Figures 8 and 9 present the variation of the two eigenvalues in low (0% to 25%) and mid (45% to 65%) power classes. The power de-rating is clearly present in both eigenvalues in figure 9, whereas no change is seen for the low power class (figure 8). These observations lead to the the conclusion that the performance is influenced only in certain power bins and thus the most probable root cause is a deliberate control action by the turbine operator. The result from a vibration-based condition monitoring point of view is positive step changes on the gearbox speed related measurements during these periods. The latter can be considered as a sign of sudden changes in the drive train dynamics denoting a faulty operation of one or more components. Hence the eigenvalue trending can be used to detect any changes in the performance of the turbine which coincide with changes in the vibration data.

Two different control actions have caused power production variations on Turbine#07. Firstly, the power was de-rated to 1/3 of P_n for a short period of time in mid 2012. This action yielded changes to both eigenevalues as it was seen for Turbine#09 in figure 9. Then, a noise reduction mode was enabled for the current wind turbine (and for the vast majority of the turbines in the park) many times in 2012 and 2013. The noise reduction mode corresponds to the mitigation of the aerodynamic noise emitted by the blades by reducing the

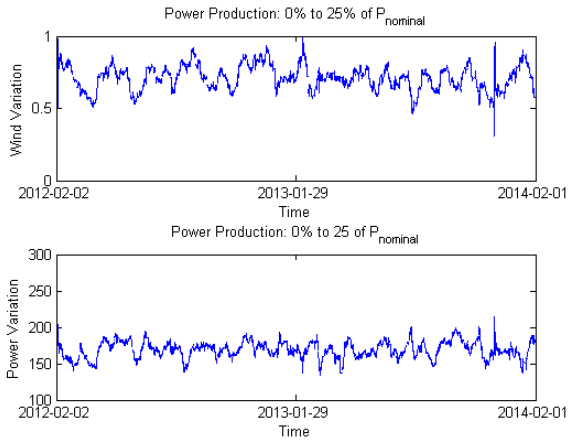


Figure 8. Turbine#09 - Power de-rating - Trending behaviour of eigenvalues in low power production.

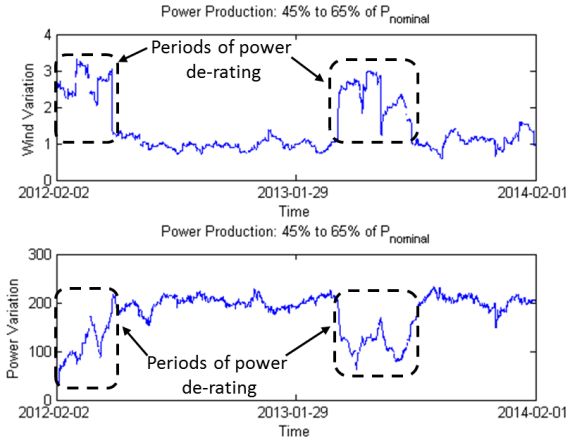


Figure 9. Turbine#09 - Power de-rating - Trending behaviour of eigenvalues in mid power production.

main rotor running speed. In this case, only the wind variation subplot presents increased trends matching the periods where this operation mode was active. It can be remarked that the wind and power variation is not affected by the operational changes in low power production. Thus, as for Turbine#09, the fact that the trends of the low power bin are stables indicates that the most likely origin of the increase in mid power production is again due to an intentional control action.

At this point, it is important to emphasize that the recognition of the power generation changes is solely based on the comparison between the normal behaviour and any decrease or increase of either the wind or power variation trends. This approach excludes the dependency from the power curve provided by the manufacturer. In addition, any site related factors influencing the power output profile of the turbine under investigation are implicitly included.

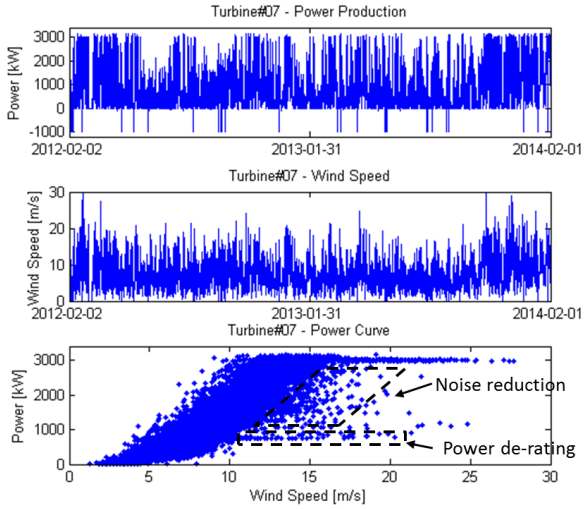


Figure 10. Turbine#07 - Power Production, Wind Speed and Power Curve - Case: Enabling of noise reduction mode.

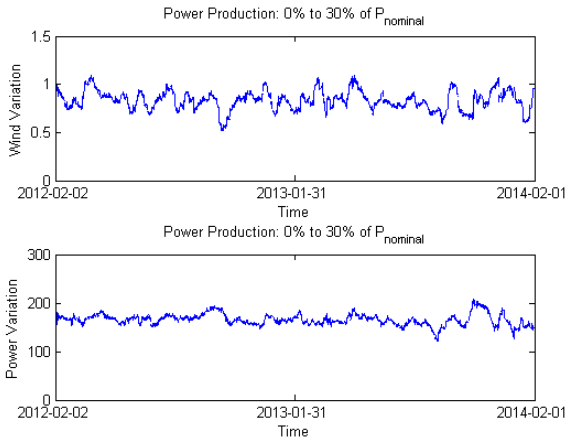


Figure 11. Turbine#07 - Noise reduction mode - Trending behaviour of eigenvalues in low power production.

5. DISCUSSION

The analysis presented in the previous sections attempted to illustrate the condition monitoring capabilities of the power performance technique. As condition monitoring systems rely on alarms when an alert or danger limit is violated, the same approach can be adopted in this case as well. The authors of the present paper are currently working on setting customized alert limits for each turbine individually after a short learning period (approximately one month) and global danger limits for each turbine type.

The results of the power performance monitoring method can be also applicable to other functions related to the operation of the turbine. A potential application is the enabling of de-icing systems installed in turbines erected in cold climate lo-

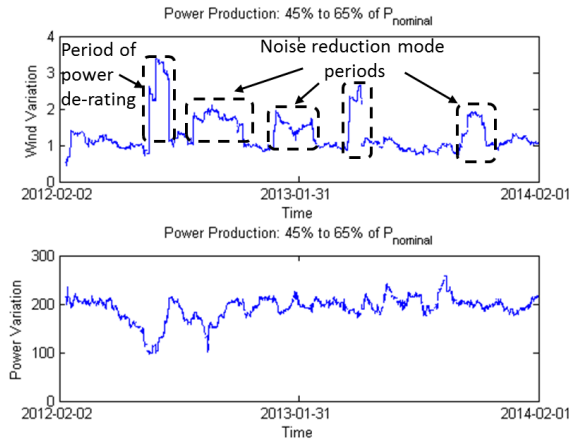


Figure 12. Turbine#07 - Trending behavior of eigenvalues in mid power production.

cations. By combining indications from the power performance analysis technique and the ambient temperature, the de-icing systems can be triggered in order to avoid long offline periods by consuming a portion of the energy production for heating the blades and the nacelle.

6. CONCLUSIONS

In this paper, changes in eigenvalues of wind speed - power production data sets are employed as power performance monitoring tools. Three cases have been analysed and presented: icing, power de-rating and noise reduction mode. The analysis has shown that detection of power production abnormalities can be achieved without necessity of the power curve provided by the turbine manufacturer, but based solely on the trending behaviour of the two eigenvectors. Furthermore, the division of the power output into discrete power classes has provided essential information regarding the identification of the most likely root cause of the power generation change. Finally, with high time resolution of the field data, the presented approach adds value to existing diagnostics, based on vibration, resulting in a comprehensive evaluation of the turbine state and consistent identification of issues during operation.

REFERENCES

Andersson, C., Gutt, S., & Hastings, M. (2007). Cost–effective monitoring solution using external surveillance centre. In *The 2nd world congress on engineering asset management (eam) and the 4th international conference on condition monitoring.*

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer New York.

Butler, S., Ringwood, J., & O'Connor, F. (2013). Exploiting scada system data for wind turbine performance monitoring. In *Control and fault-tolerant systems (systol),*

2013 conference on.

Cios, K. J., Pedrycz, W., Swiniarski, R. W., & Kurgan, L. A. (2007). *Data mining: A knowledge discovery approach*. Springer.

Lu, B., Li, Y., Wu, X., & Yang, Z. (2009). A review of recent advances in wind turbine condition monitoring and fault diagnosis. In *Power electronics and machines in wind applications* (pp. 1–7).

Lydia, M., Selvakumar, A. I., Kumar, S. S., & Kumar, G. E. P. (2013). Advanced algorithms for wind turbine power curve modeling. *IEEE Transactions on Sustainable Energy*, *4*, 827–835.

Marhadi, K., & Hilmisson, R. (2013, June). Simple and effective technique for early detection of rolling element bearing fault: A case study in wind turbine application. In *International congress of condition monitoring and diagnostic engineering management* (pp. 94–97).

Mchali, M., Barthelmie, R. J., Frandsen, S. T., Jensen, L. E., & Rthor, P.-E. (2006). Wake effects at Horns Rev and their influence on energy production. In *European wind energy conference.*

Park, J.-Y., Lee, J.-K., Oh, K.-Y., & Lee, J.-S. (2014). Development of a novel power curve monitoring method for wind turbines and its field tests. *IEEE Transactions on Energy Conversion*.

Schlechtingen, M., Santos, I. F., & Achiche, S. (2013). Using data-mining apapproach for wind turbine power curve monitoring: A comparative study. *IEEE Transactions on Sustainable Energy*, *4*, 671–679.

Uluyol, O., Parthasarathy, G., Foslien, W., & Kim, K. (2011). Power curve analytic for wind turbine performance monitoring and prognostics. In *Annual conference of the prognostics and health management society* (Vol. 20).

BIOGRAPHIES

Georgios Alexandros Skrimpas was born in Athens, Greece in 1986. He received the Diploma in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2009 and the M. Sc. in wind energy from the Technical University of Denmark (DTU) in 2012. He joined Brüel and Kjær Vibro in 2012 and since 2013 he is pursuing the Industrial Ph.D. degree at the Centre of Electric Power and Energy at DTU in cooperation with Brüel and Kjær Vibro. His research interests are diagnosis and prognosis of electrical and mechanical faults in wind turbines.

Christian Walsted Sweeney was born in Copenhagen, Denmark in 1982. He received the B.Sc. from the University of Southern Denmark in 2006 and the M.Sc from the Technical University of Denmark in 2008 both in mechanical engineering. From 2008 to 2010 he was employed as a diagnostic engineer and since 2010 he is the team leader of the diagnostic services group. His research focus is on the development of condition monitoring systems and handling of large data quantities.

Kun S. Marhadi is an engineer in the Remote Monitoring Group at Brüel and Kjær Vibro. He joined Brüel and Kjær Vibro in 2012. Previously, he worked as a postdoctoral fellow in the Department of Mathematics at the Technical University of Denmark (DTU). He received his PhD in computational science in 2010 from San Diego State University and Claremont Graduate University. He has M.S. and B.S. in aerospace engineering from Texas A&M University. His expertise is in structural vibration and analyses, probabilistic methods, and design optimization.

Bogi Bech Jensen received the Ph.D. degree from Newcastle University, Newcastle Upon Tyne, U.K., for his work on induction machine design. He was in various engineering and academic positions in the marine sector from 1994 to 2004. He was at Newcastle University from 2004 to 2010 first as a Postgraduate, then Research Associate and finally as a Lecturer. From 2010 to 2014 he was Associate Professor and later Head of Research Group at the Technical University of Denmark (DTU), Lyngby, Denmark. He is currently Professor of Energy Engineering at the University of the Faroe Islands (UFI), where he is responsible for education and research in energy.

Nenad Mijatovic received his Ph.D. degree from the Technical University of Denmark for his work in superconducting machine. After obtainnign his Dipl.Ing. education at University of Belgrade, Serbia, he enrolled as a doctoral candidate in 2012. Upon completion of the PhD, he has continued to work in the same field of machine research - superconducting machines, as an Industrial PostDoc. The 3 year industrial PostDoc grant has been provided by Hojteknologifonden and supported by Envision Energy Aps., Denmark. Dr. N. Mijatovic is a member of IEEE from 2008 and his field of interest and research includes novel electrical machine design, operations and diagnostic.

Joachim Holbøll is associate professor and deputy head of center at DTU, Department of Electrical Engineering, Center for Electric Power and Energy. His main field of research is high voltage components, their properties, condition and broad band performance, including insulation systems performance under AC, DC and transients. Focus is also on wind turbine technology and future power grid applications of components. J. Holbøll is Senior Member of IEEE.

A PHM Approach to Additive Manufacturing Equipment Health Monitoring, Fault Diagnosis, and Quality Control

Jae Yoon[1], David He[2], and Brandon Van Hecke[3]

[1,2,3]*University of Illinois at Chicago, Chicago, IL, 60607, USA*
jyoon52@uic.edu
davidhe@uic.edu
bvanhe2@uic.edu

ABSTRACT

Fabrication of three-dimensional (3D) objects through direct deposition of functional materials using 3D printing equipment is called additive manufacturing (AM). Benefits of AM include producing goods quickly and on-demand, with greater customization and complexity and less material waste. While the use of AM has been growing, a number of challenges continue to impede its more widespread adoption, particularly in the areas of non-destructive evaluation/non-destructive testing (NDE/NDT) techniques for AM equipment health monitoring and measurement. In this paper, a prognostics and health management (PHM) approach to AM equipment health monitoring, fault diagnosis and quality control is presented and illustrated with a case study. The presented PHM approach is developed using two types of NDE/NDT sensors: acoustic emission (AE) sensor and piezoelectric strain sensor. A seeded driving belt fault on a fused filament fabrication desktop 3D printer is used to validate the feasibility of the PHM approach in the case study. The case study results have shown the effectiveness of the presented method for AM equipment fault diagnosis and quality control.

1. INTRODUCTION

In his 2013 state of the union address, US President Obama called three-dimensional (3D) printing "the potential to revolutionize the way we make almost everything" (Office of the Press Secretary, 2013). Fabrication of 3D objects through direct deposition of functional materials using 3D printing equipment is called additive manufacturing (AM). Benefits of AM include producing goods quickly and on-demand, with greater customization and complexity and less material waste. If the modern manufacturing which was subtractive process by cutting or milling is optimized at mass production, the future manufacturing would be called a creative customization through 3D printing at consumers' will.

While the use of AM has been growing, a number of challenges continue to impede its more widespread adoption, particularly in the areas of non-destructive evaluation/non-destructive testing (NDE/NDT) techniques for AM equipment health monitoring and measurement. According to a recent report on measurement science roadmap for metal-based additive manufacturing (Energetics Incorporated, 2013), current technical barriers or challenges in AM were roughly categorized as materials, process and equipment, qualification and certification, and modeling and simulation. Particularly in the process and equipment category, the highest priority in NDE/NDT techniques have been specified as: (1) Combining NDE techniques to better assess quality via an integrated approach; (2) Adapting existing NDE techniques to AM, especially parts, and characterizing defects; (3) Lack of affordable quality inspection tools for direct metal parts. Even though the 3D printing technology has been available since 80s, it was not until recent days that 3D printing came to the fore in commercial manufacturing. Thus, very few studies have been conducted on NDE based 3D printer health monitoring and prognostics. The AM has two unique characteristics: (1) relatively long cycle time; (2) high quality standard for dimension accuracy. These unique characteristics of AM can be considered as good opportunities for developing PHM based approach for 3D printer health monitoring, fault detection and quality control. As the dimension accuracy of the printed product can be caused by inaccurate movement of the 3D printer, by detecting the 3D printer fault and stopping the faulty execution of the printing process, manufacturing time, materials, and cost can be saved and product quality assured.

In the related field of rotating machinery fault detection and diagnostics, the use of different NDE/NDT techniques such as acoustic emission (Yoshioka and Fujiwara, 1984; Tandon and Mata, 1999; Tandon and Narka, 2000; Scheer *et al*.,

Jae Yoon *et al*. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2007; Bechhoefer *et al.*, 2013; Qu *et al*., 2013 and 2014), torsional vibration (Feng & Zuo, 2013), and fiber optic strain sensors (Kiddy *et al*., 2011) has been investigated with drivetrain in wind turbine and rotorcraft. In this study, the potential capability of acoustic emission (AE) and piezoelectric (PE) strain sensors as fault detection and quality control technique for AM equipment and product is investigated.

AE is commonly defined as transient elastic waves within a material, caused by the release of localized stress energy (Mathews, 1983). The advantage of using AE sensor as failure analysis source is that AE propagates from the epicenter to sensing apparatus within materials while vibration sensor requires perpendicular installation along with the vibration direction. Identifying vibration direction is sometimes painful if their sources are combinative. Also, AE signals are distinguishable from acoustic signals in that acoustic signals generally lie on the audible range of human (*e.g.* 20 Hz ~ 20 kHz). On the other hand, AE signals lie on a higher frequency range (*e.g.* 1 kHz ~ 1 MHz). Thus a high sampling rate between 2 to 10 MHz has been a typical choice of sampling rate for AE data collection. Other issues may arise including a high data volume and complicated feature of AE signals, which make the AE data processing challenging. However, it has been also reported that AE sensors are more sensitive in early fault detection than vibration sensors with various gear and bearing fault diagnostic applications (Yoshioka and Fujiwara, 1984; Tandon and Mata, 1999; Tandon and Narka, 2000; and Scheer *et al*., 2007).

The feasibility of using fiber optic strain sensors to detect damaged gearbox was recently reported by Kiddy *et al*. (2011). In their study, fiber optic strain sensors were mounted on a helicopter transmission test rig to investigate the detectability of gear fault conditions. However, the low maximum sampling rate (up to 1 kHz) of the fiber optic strain sensor limits its wide applicability in machinery fault detection. On the other hand, the PE strain sensors measure torsional vibration by quantifying terminal voltage difference released by deformed piezoelectric material. Unlike the fiber optic strain sensor, PE strain sensor has a merit in higher sampling rate up to 100 kHz. Compared to the conventional strain gauge sensors and accelerometers, the PE strain sensors have certain advantages that could be summarized as follows: (1) ability to measure the first derivative of physical deformation, (2) high linearity and sensitivity from their superior noise immunity as compared to differentiated sensing performance of conventional strain sensors (Lee and O'Sullivan, 1991; Banaszak 2001), (3) high frequency range (Jiang *et al.*, 2014), (4) space-efficiency without a structural change on the measuring target (Kon *et al*., 2007), and (5) negligible temperature effect on the measurement output (Sirohi and Chopra, 2000; Jiang *et al.*, 2014). The aforementioned benefits allow PE strain sensors to potentially have greater sensing resolution and accuracy.

Up to today, no investigation on 3D printer health monitoring and fault diagnosis has been reported in the literature. In this paper, an investigation into the feasibility of PHM based AE and PE strain signal analysis techniques for 3D printer fault detection and quality control is reported. The remainder of the paper is organized as follows. Section 2 provides a detailed explanation of the proposed methodology. In Section 3, the details of the experimental setup and the seeded fault tests on a 3D printer test rig for validating the proposed methodology are provided. Section 4 presents the 3D printer fault detection results from the seeded fault tests. Finally, Section 5 concludes the paper.

2. METHODOLOGY

An overview of the proposed methodology is provided in Figure 1. As shown in Figure 1, a data acquisition (DAQ) system is used to collect the AE signals and PE strain signals at the same time. While the PE sensor is directly connected to the DAQ, the AE sensor, on the other hands, is connected to the DAQ board through a hardware based heterodyne frequency reduction device. Then, filter bands are chosen for each sensor to remove noise in the collected signals before they can be used to compute condition indicators (CIs) for fault detection. The key components of the methodology are explained in the next two sections. Section 2.1 provides a brief review of the hardware based heterodyne technique for AE sensor and the computation of CIs for 3D printer fault detection is followed in Section 2.2.

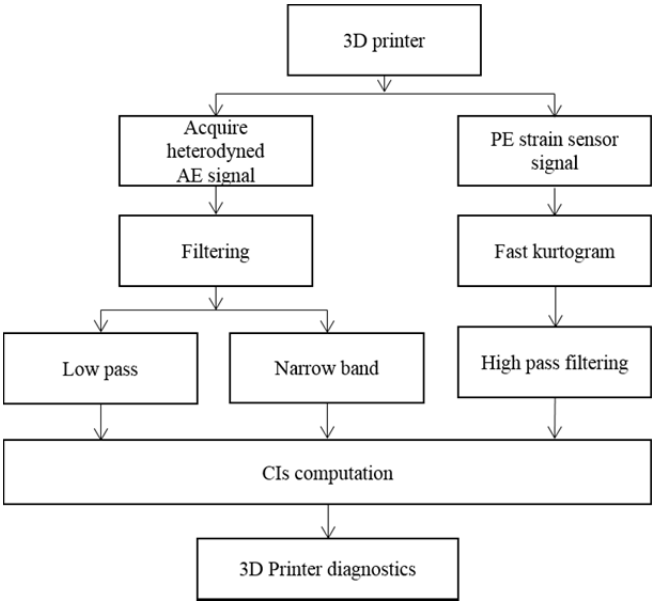


Figure 1. Overview of the 3D printer fault diagnosis with PE strain sensor and AE sensor.

2.1. The Heterodyne Technique

To apply AE based NDE/NDT techniques to machine fault detection and diagnosis, one technical challenge is to deal with the data storage and processing burden caused by the typical high sampling rate of AE sensor (from several MHz to 10 MHz). To meet the challenge, frequency shifting technique, namely heterodyne (Fessenden, 1913) based AE fault detection and diagnosis methods have been developed for gearboxes (Bechhoefer *et al.*, 2013; Qu *et al*., 2013; 2014). The heterodyne technique downshifts the frequency of the AE signals so that a sampling rate comparable to vibration analysis can be utilized. Qu *et al*. (2013 and 2014) have shown the effectiveness AE based fault detection and diagnosis using heterodyne technique with a sampling rate as low as to 20 kHz for a split torque type gearbox. The AE based NDE/NDT techniques implemented with heterodyne are significant as size of AE data needs to be stored and the computational cost can be significantly reduced. The heterodyned AE technique implemented in this paper works similarly to a radio quadrature demodulator: shifting the carrier frequency to baseband, followed by low pass filtering. Mathematically, heterodyning is based on the trigonometric identity. For two signals with different frequency f_1 and f_2, respectively, their product could be written as:

$$\sin(2\pi f_1 t)\ \sin(2\pi f_2 t) = \frac{1}{2}\cos[2\pi(f_1 - f_2)] - \frac{1}{2}\cos[2\pi(f_1 + f_2)] \quad (1)$$

where f_1 is the AE carrier frequency and f_2 is the demodulator's reference signal frequency. In applications, any desired new output signals called as heterodynes, one at the sum $f_1 + f_2$, and the other at the difference $f_1 - f_2$, are utilized upon necessity. Technically, the heterodyning technique is aimed especially at demodulating the amplitude modulated signals. The amplitude modulation process can be mathematically expressed as:

$$U_a = (U_m + mx)\cos\omega_c t \quad (2)$$

where, U_a is the amplitude modulated signal, U_m is the carrier signal amplitude, m is the modulation coefficient, x is the signal of interest, and ω_c is the carrier signal frequency. By introducing an amplitude and frequency for x by X_m and Ω, respectively, the signal of interest x can be represented as:

$$x = X_m cos\Omega t \quad (3)$$

Note that it is assumed that Ω is much smaller than ω_c. Then, with the heterodyning technique, the modulated signal will be multiplied by a unit amplitude reference signal $\cos(\omega_c t)$. Then the resulting U_o can be written as:

$$U_o = (U_m + mx)\cos(\omega_c t)\cos(\omega_c t) = (U_m + mx)\left[\frac{1}{2} + \frac{1}{2}\cos(2\omega_c t)\right] \quad (4)$$

Substituting Eq. (3) into Eq. (4) yields:

$$U_o = \frac{1}{2}U_m + \frac{1}{2}mX_m cos\Omega t + \frac{1}{2}U_m cos(2\omega_c t) + \frac{1}{4}mX_m[cos(2\omega_c + \Omega)t + cos(2\omega_c - \Omega)t] \quad (5)$$

Since U_m is assumed not to contain any useful information related to the modulated signal, it could be canceled out. From Eq. (5), it can be concluded that only the second term $\frac{1}{2}mX_m cos\Omega t$ will remain after applying low pass filter, while the high frequency components around frequency $2\omega_c$ will be removed. In the final heterodyning demodulation step, the signal frequency can be reduced to 10s of kHz. The resulting frequency range for AE signals becomes comparable to that of typical vibration signals. Thus, a lower sampling rate in an AE data acquisition system can be used. The heterodyned AE data acquisition procedure is shown by comparing it with the conventional AE method in Figure 2.

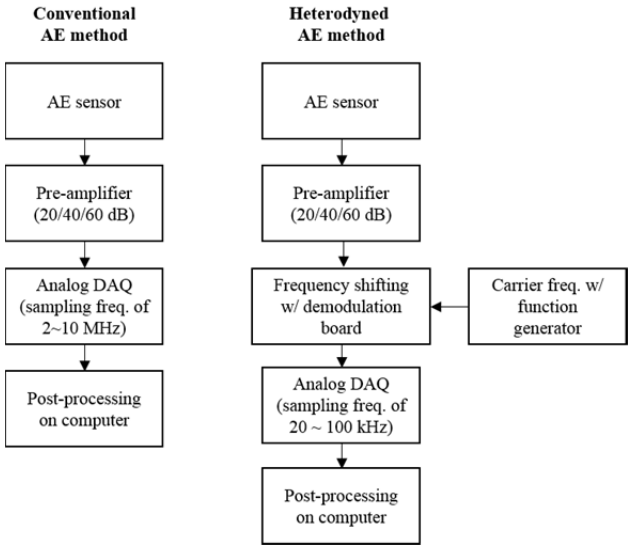


Figure 2. Comparison of the heterodyned AE data acquisition procedure with the conventional AE methods.

Finding a proper reference signal is critical to the successful implementation of the heterodyne technique in AE data acquisition. Since each AE sensor product from varying manufacturers has a unique frequency characteristic, the following optimization process based a linear chirp function is performed so that the root mean square (RMS) of the demodulated output signal could be maximized. The optimization process is described in Qu *et al*. (2014).

2.2. CIs for 3D Printer Fault Detection

Table 1 provides the definitions of CIs investigated for 3D printer fault detection in this paper. The CIs can be defined into five general types: root mean square (*RMS*), peak to

Table 1. The definitions of the CIs.

		Input Signal (x_{IN})				
		Raw AE	EO	NB	AM	FM
CI	Description / Equation	Raw heterodyned AE data (x_{raw})	Energy operator: a residual of the autocorrelation function (x_{EO})	Narrow band pass filtered (x_{NB})	Amplitude modulation of NB filtered signal ($AM(x_{NB})$)	Frequency modulation of NB filtered signal ($FM(x_{NB})$)
Root mean square (*RMS*)	$RMS(x_{IN}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	$RMS(x_{IN})$: measures the magnitude of a discretized signal.				
Peak to peak (*P2P*)	$P2P(x_{IN}) = \frac{(\max(x_{IN}) - \min(x_{IN}))}{2}$	$P2P(x_{IN})$: measures the maximum difference within the data range.				
Skewness (*SK*)	$SK(x_{IN}) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^3}{\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\right]^3}$	$SK(x_{IN})$: measures the asymmetry of the data about its mean value. A negative *SK* value and positive *SK* value imply the data has a longer or fatter left tail and the data has a longer or fatter right tail, respectively.				
Kurtosis (*KT*)	$KT(x_{IN}) = \frac{N\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left[\sum_{i=1}^{N}(x_i - \bar{x})^2\right]^2}$	$KT(x_{IN})$: measures the peakedness, smoothness, and the heaviness of tail in a data set.				
Crest factor (*CF*)	$CF(x_{IN}) = \frac{P2P(x_{IN})}{RMS(x_{IN})}$	$CF(x_{IN})$: measures the ratio between $P2P(x_{IN})$ and $RMS(x_{IN})$ to describe how extreme the peaks are in a waveform.				

Note: x_i is i^{th} element of the input data x_{IN}; N is the length of the input data x_{IN}; max(·) returns the maximal element of input data x_{IN}; min(·) returns the minimal element of input data x_{IN}; $\bar{x}$ is a mean value of the input data x_{IN} defined as $\sum_{i=1}^{N} x_i / N$

peak (*P2P*), skewness (*SK*), kurtosis (*KT*), and crest factor (*CF*). Each type of CI can be computed using different input signals. In addition to raw signals, other types of input signals can be generated: energy operator (EO), narrow band (NB), AM, and FM. The EO introduced by Teager (1992) is defined as the residual of the autocorrelation function as following:

$$x_{EO,i} = x_i^2 - x_{i-1} \cdot x_{i+1}, \quad (\text{for } i = 2, 3, \dots, N-1) \tag{6}$$

where $x_{EO,i}$ is the i^{th} element of EO data; x_i is the i^{th} element of the input data x_{IN}. NB is the time domain representation after applying narrow band of interest which could be seen in frequency domain. Finally, AM and FM are obtained by amplitude modulation and phase modulation of the NB filtered data.

3. EXPERIMENTAL SETUP

This section covers the experimental setup used to establish the AE and PE strain sensor based 3D printer fault diagnosis technique. The methodologies were validated with a desktop 3D printer using fused filament fabrication. Section 3.1 introduces the 3D printer test rig and Section 3.2 covers the seeded fault test.

3.1. The 3D Printer Test Rig

Figure 3 shows the 3D printer test rig and the DAQ system used in the seeded fault test in this paper.

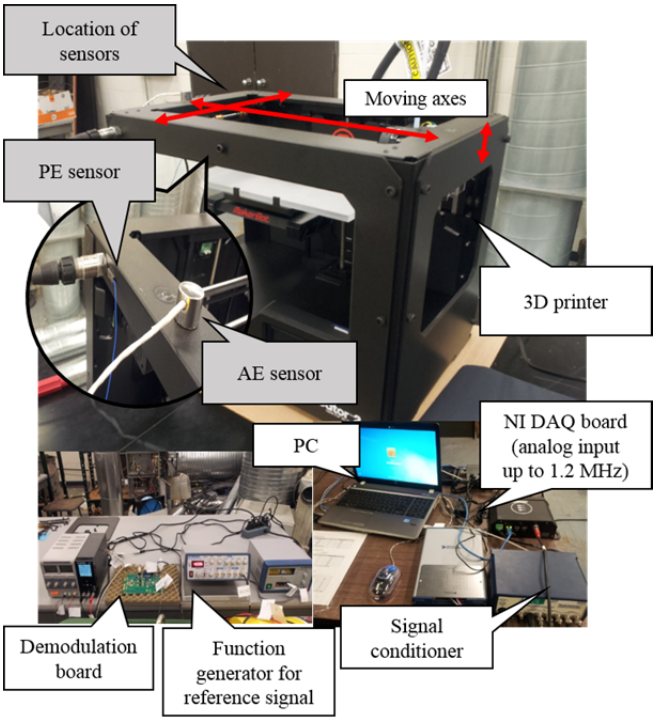


Figure 3. The 3D printer test rig and the DAQ system.

The 3D printer test rig composes two main parts: (1) heterodyned AE based DAQ system, (2) 3D printer. The DAQ system includes a National Instruments' DAQ board with a maximum analog input sampling rate of 1.25 MHz, AE sensor attached on the 3D printer, demodulation board (AD8339), analog amplifier with gain 20/40/60dB, and function generator. The 3D printer (Makerbot, 2014) has a layer resolution up to100 µm, position precision of 11 µm on *X* and *Y* axes and 2.5 µm on *Z* axis, and a nozzle of 0.4 mm diameter controlled by two stepper motors and wear resistant oil-infused bronze bearings.

3.2. 3D Printer Seeded Fault

According to the troubleshooting maintenance document (Makerbot, 2014) of the machine, one potential problem is the looseness of the belt driving the motion of the extruder nozzle. Thus, a malfunctioned toothed belt scenario was artificially created and simulated in this paper. The seeded fault was created by inserting five small pieces of metal wire into the slots between teeth of belt to create faulty operation during printing process. Figure 4 shows the seeded fault created by inserting a metal wire piece into the slot between two teeth on the toothed driving belt to simulate the looseness of the driving belt. The inserted metal wire piece was cut into the same dimensions in size as the slot between the belt teeth so that the slot was perfectly filled with the metal wire piece. Then the metal wire piece was tied on the belt with a thin flexible tape.

Figure 4. Seeded fault on toothed belt.

The 3D printer was run with and without the fault seeded driving belt to produce ten sets of bolt and nut (five sets for each conditions). Individual run took about 28 minutes to print one set of bolt and nut. For sample consistency, a total of six heterodyned AE data samples were recorded for 10 seconds at pre-specified time locations from each run. The data acquisition procedure for the seeded fault test is depicted with a flowchart in Figure 5.

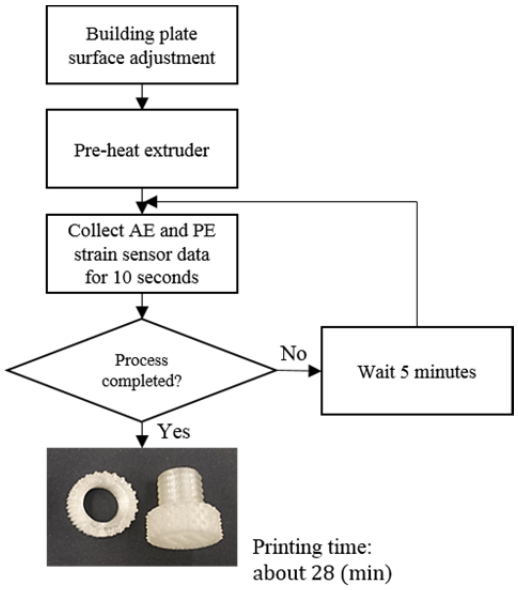


Figure 5. Data acquisition procedure.

Figure 6 shows the 3D outputs of the healthy and faulty 3D printers. Under the normal printing conditions, the printed nut and bolt should smoothly thread together and function as intended. Under the faulty printing condition, even though the pair of printed bolt and nut appears to be normal, the bolt can only be turned into the nut half way. This clearly indicates that the threads on the bolt or inside the nut

were not printed up to the required precision due to the driving belt fault in the 3D printer.

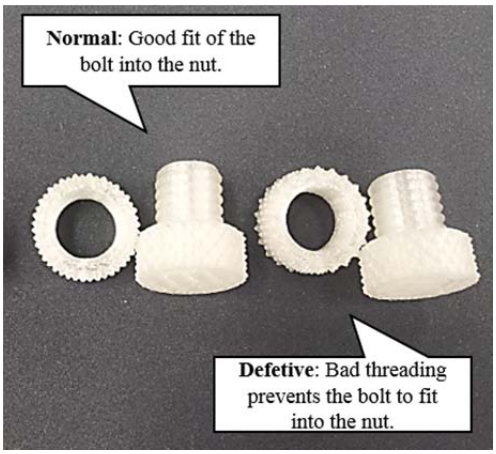


Figure 6. 3D outputs from the healthy and the faulty 3D printers.

4. RESULTS

This section covers the 3D printer fault diagnostic results from the AE and PE strain sensor based technique. Section 3.1 explains AE signal analysis results and Section 3.2 the PE strain sensor signal analysis results.

4.1. AE Signal Analysis Results

The AE signal analysis results for the seeded fault tests conducted on the 3D printer test rig are provided in this section. Figure 7 shows the spectrums of AE data samples. By examining the spectrums in Figure 7, two different frequency regions were chosen for the low pass and narrow band pass filters: low frequency region up to 20 kHz and narrow band frequency around 3906 Hz. As shown in Figure 7, a remarkably high peak was observed within low pass range from all of AE samples. These peaks are specifically located at 3906 Hz. So a narrow band pass filter with a band width of 3906±3 Hz around the peak frequency location was chosen.

In Figure 8, RMS result from the low pass filter is provided. The resulting RMS of the heterodyned AE sample showed clear separation between healthy and faulty 3D printing condition. In Figure 8(a), RMS values at each sample location and trial are presented. In Figure 8(b), the averaged RMS values with a 95% confidence interval at each sample location are provided.

In Figures 9 to 12, CIs from the narrow band filtered AE signals are provided. Among all the CIs tested, majority of those that show a clear separation between the healthy and faulty conditions were computed from narrow band filtered signals. Note that the bandwidth of this narrow band is in the low frequency filter region. A clear separation between the healthy and faulty 3D printing conditions with a 95% statistical significance can be observed for the following AE based CIs: RMS, NB-RMS, NB-P2P, AM-RMS, and AM-P2P.

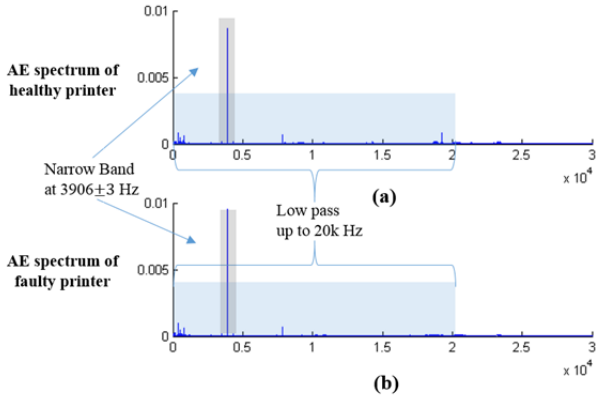


Figure 7. Spectrum of 3D printer AE signal samples: (a) healthy, (b) faulty.

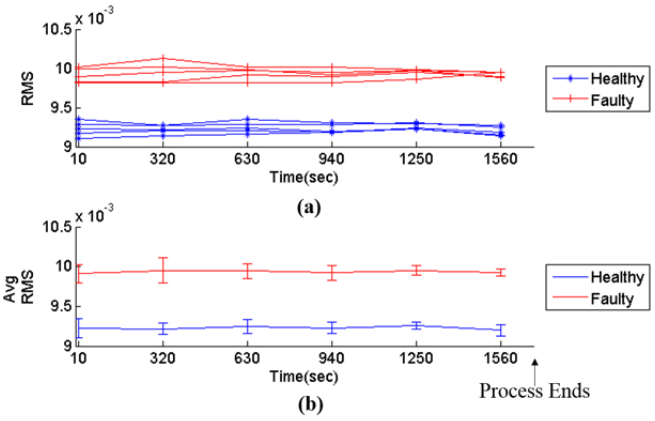


Figure 8. RMS of healthy and faulty low pass filtered results: (a) all data, (b) average with 95% confidence interval.

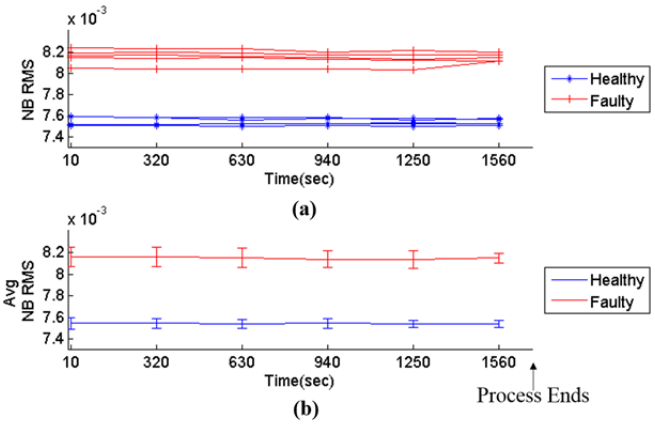


Figure 9. RMS from narrow band pass filtered result: (a) all data, (b) average with 95% confidence interval.

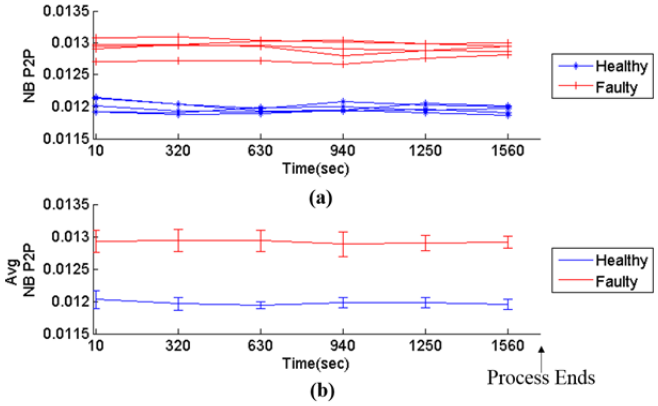


Figure 10. Peak to peak from narrow band pass filtered result: (a) all data, (b) average with 95% confidence interval.

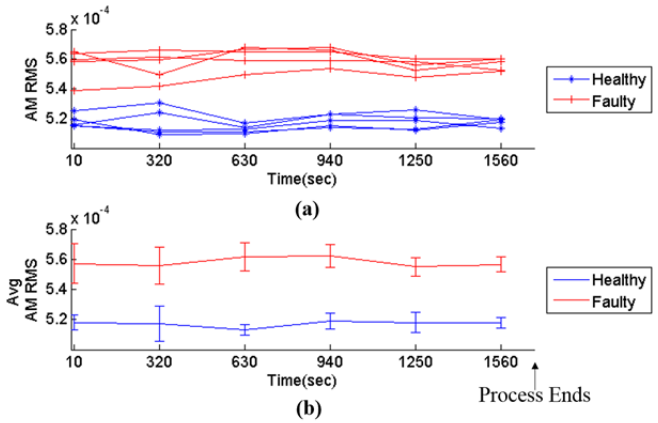


Figure 11. RMS from amplitude modulation result after narrow band pass filtered result: (a) all data, (b) average with 95% confidence interval.

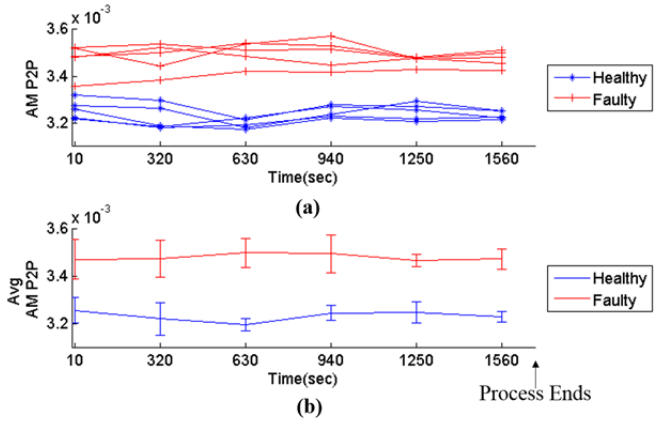


Figure 12. Peak to peak from amplitude modulation result after narrow band pass filtered result: (a) all data, (b) average with 95% confidence interval.

4.2. PE Strain Signal Analysis

In processing the PE strain sensor signals to extract CIs for the 3D printer fault detection, a similar strategy used by Kiddy *et al.* (2011) was applied. In their study, PE strain signals were divided into two parts based on their frequency: low frequency part and high frequency part. Actual damage detection was performed on the high frequency part of the strain sensor data using condition indicators. Thus, in this research, high pass filtered PE strain signals were used to compute the CIs. In search for the appropriate filter band, the fast kurtogram (Antoni, 2007) was applied to exam the impulsivity locations of PE strain signals collected from the healthy 3D printers.

Provided in Figure 13, a sample fast kurtogram result from the healthy 3D printer is displayed. The area in dark red color indicates the location of impulsivity. Statistical result of the fast kurtogram is summarized in Table 2. The 90% and 95% trimmed mean indicate that the impulsivity of PE sensor signals are located around 3.3 kHz to 4.2 kHz, respectively. Thus, a high pass band above 3 kHz was selected. Here a X % trimmed mean is the average of the data after $(100 - X)$% of the outliers are removed.

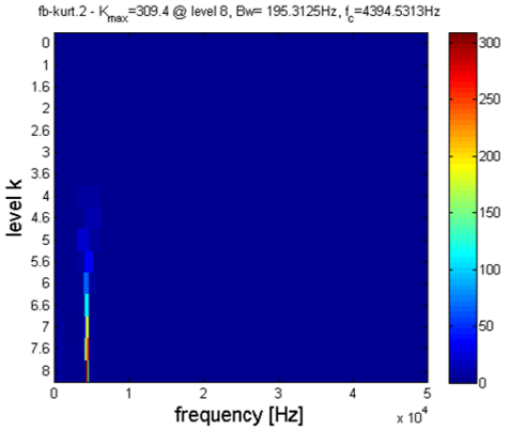


Figure 13. A sample fast kurtogram of PE strain sensor result from the healthy 3D printer.

Among all the CIs computed using PE strain signals, only RMS showed a clear separation of the faulty condition from the normal condition. The RMS of the PE stain signals and the averaged RMS with 95% confidence intervals for both the healthy and faulty conditions are provided in Figure 14.

Table 2. Statistical results of Fast kurtogram.

	Healthy	
	90% trimmed mean	95% trimmed mean
Center frequency value (Hz)	3320	4199

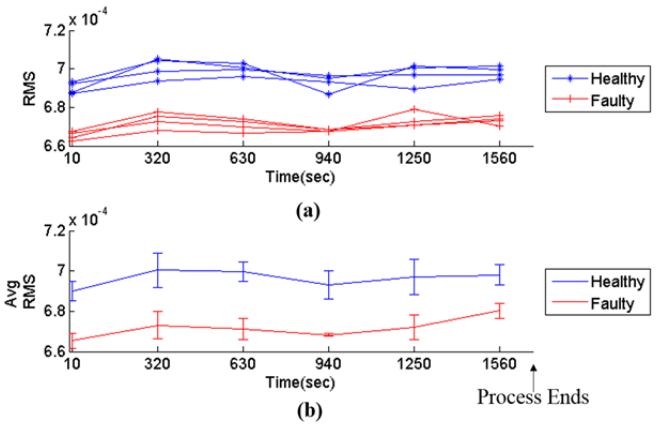


Figure 14. RMS of healthy and faulty PE strain signals: (a) all data, (b) average with 95% confidence interval.

4.3. Results Summary

The 3D printer seeded fault detection results using both the AE sensor and PE strain sensor can be summarized in Table 3.

Table 3. Summary of the 3D printer fault detection results using AE and PE strain sensors.

Sensor Type	AE Sensor		PE Strain Sensor
Sampling frequency	100 kHz		100 kHz
Filter bandwidth	Low pass band (< 20 kHz)	Narrow band (3906 ± 3 Hz)	High pass band (> 3k Hz)
Effective CIs selected	RMS	NB-RMS, NB-P2P, AM-RMS, AM-P2P	RMS

As shown in Table 2, for both the AE sensor and PE strain sensor used in the case study for 3D printer fault detection, a sampling rate of 100 kHz was used for data acquisition. For AE sensor signals, two band pass filters were used: a low pass filter and a narrow band filter. When the low pass filter was used, RMS provided the best performance and was able to detect the fault. When a narrow band filter was used, the following CIs were able to detect the fault: NB-RMS, NB-P2P, AM-RMS, and AM-P2P. For PE strain sensor signals, a high pass filter was used and only one CI, RMS, was able to detect the fault.

It has to be pointed out that the 3D printer fault was detected right after the first sample data was collected by both the AE and PE strain sensor in the seeded fault test. In real AM application, it can take up to several days to print out a product by a 3D printer. Therefore, significant amount of manufacturing time, materials, and cost can be saved and the quality of the product can be assured if a 3D printer fault can be detected and the printer be stopped days before it finishes printing the defective product.

5. CONCLUSIONS

In this paper, an investigation into the feasibility of PHM based AE and PE strain signal analysis techniques for 3D printer fault detection and quality control was reported. The presented PHM approach was developed using two types of NDE/NDT sensors: AE sensor and piezoelectric strain sensor. A seeded driving belt fault on a fused filament fabrication desktop 3D printer was used to validate the feasibility of the PHM approach in the case study. For the AE signal analysis in particular, a high peak in the frequency domain was detected and a narrow band pass filter around the peak was used to extract multiple condition indicators to detect the fault. On the other hand, in the PE strain analysis, the fast kurtogram was used to determine the proper high-pass filter band to obtain high frequency components to obtain effective fault detection CIs. The results have shown that the driving belt seeded looseness fault could be detected by both of the AE and PE strain sensor analysis methods. The methods presented in this paper could be extended to other potential 3D printing faults such as material feed or additional mechanical component faults.

REFERENCES

Antoni, J. (2007). Fast computation of the kurtogram for the detection of transient faults, *Mechanical Systems and Signal Processing*, Vol. 32, No. 1, pp. 108 - 124.

Banaszak, D. (2001). Comparison of piezoelectric strain sensors with strain gages, *Proceedings of the Annual Meeting of the American Statistical Association*, August 5 – 9, Atlanta, GA.

Bechhoefer, E., Qu, Y., Zhu, J., & He, D. (2013). Signal processing technique to improve an acoustic emission sensors, *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2013*, October 14 – 17, New Orleans, LA.

Energetics Incorporated. (2013). Workshop summary report: measurement science roadmap for metal-based additive manufacturing, *National Institute for Standards and Technology (NIST)*, U.S. Department of Commerce, May 13.

Feng, Z. & Zuo, M. J. (2013). Fault diagnosis of planetary gearboxes via torsional vibration signal analysis, *Mechanical Systems and Signal Processing*, Vol. 36, No.2, pp. 401 - 421.

Fessenden, R. A. (1913). *Electric Signaling Apparatus*, US Patent 1050441A, January 14.

Lee, C. K. & O'Sullivan, T. (1991). Piezoelectric strain rate gages," *Journal of the Acoustical Society of America*, Vol. 90, No.2, pp.945 - 953.

Kiddy, J. S., Samuel, P. D., Lewicki, D. G., LaBerge, K. E., Ehinger, R. T., & Fetty, J. (2011). Fiber optic strain sensor for planetary gear diagnostics, *NASA Technical Report: NASA/TM-2011-217123*, NASA Glenn Research Center, Cleveland, OH.

Kon, S., Oldham, K., & Horowitz, R. (2007). Piezoresistive and piezoelectric MEMS strain sensors for vibration detection, *Proceedings of the SPIE: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 6529, pp. 1 - 11.

MakerBot Industries, LLC. (2014). *Replicator2 Desktop 3D Printer User Manual*, Brooklyn, NY.

Mathews, J. R. (1983). *Acoustic Emission*, Gordon and Breach Science Publishers Inc., New York, NY, USA.

Office of the Press Secretary. (2013). *Remarks by the President in the State of the Union Address*, February 12, 2013. http://www.whitehouse.gov/the-press-office/2013/02/12/remarks-president-state-union-address

Scheer, C., Reimche, W., & Bach, F.W. (2007). Early fault detection at gear units by acoustic emission and wavelet analysis, *Journal of Acoustic Emission*, Vol. 25, No. 1, pp. 331 – 340.

Sirohi, J. & Chopra, I. (2000). Fundamental understanding of piezoelectric strain sensors, *Proceedings of Journal of Intelligent Material Systems and Structures*, Vol. 11, No. 4, pp. 246 - 257.

Tandon, N. & Mata, S. (1999). Detection of defects in gears by acoustic emission measurements, *Journal of Acoustic Emission*, Vol. 17, No. 1-2, pp. 23 - 27.

Tandon, N. & Nakra, B. C. (2000). Comparison of vibration and acoustic measurement techniques for the condition monitoring of rolling element bearings, *Tribology International*, Vol. 25, No. 3, pp. 205 - 212, 1992.

Teager, H. M., & Teager S. M. (1992). Evidence for nonlinear sound production mechanisms in the vocal tract, , *Speech Production and Speech Modeling Symposium, Time Frequency and Time-Scale Analysis*, edited by Hardcastle, W. J. and Marchal, A., Springer, Amsterdam, Netherlands.

Qu, Y., He, D., Bechhoefer, E., & Zhu, J. (2013). A new acoustic emission sensor based gear fault detection approach, *International Journal of Prognostics and Health Management*, Vol. 4, Special Issue on Wind Turbine PHM, pp. 1 - 14.

Qu, Y., He, D., Yoon, J., VanHecke, B., Bechhoefer, E., & Zhu, J. (2014). Gearbox tooth cut fault diagnostics using acoustic emission and vibration sensors - a comparative study, *Sensors*, Vol. 14, No. 1, pp. 1372 - 1393.

Yoshioka, T., & Fujiwara, T. (1984). Application of acoustic emission technique to detection of rolling bearing failure, *American Society of Mechanical Engineers*, Vol. 14, No. 1, pp. 55 - 76.

BIOGRAPHIES

Jae Yoon received his B.E degree in Control engineering from Kwangwoon University, Republic of Korea, worked at Samsung Electronics Co. Ltd. as a product engineer from 2006 through 2008. He then received M.S. degree in Mechanical engineering from the University of Florida. He joined the Intelligent Systems Modeling & Development Laboratory in the department of Mechanical and Industrial Engineering at the University of Illinois-Chicago to pursue Ph.D. degree. His current research interests include: machinery health monitoring for CBM, data-driven methods for diagnostics, and model based and data mining based prognostics, encompassing reliability engineering.

David He Dr. He is a Professor and Director of the Intelligent Systems Modeling & Development Laboratory in the Department of Mechanical and Industrial Engineering at The University of Illinois-Chicago. Dr. He's research areas include: machinery health monitoring, diagnosis and prognosis, complex systems failure analysis, quality and reliability engineering, and manufacturing systems design, modeling, scheduling and planning.

Brandon Van Hecke received his B.S. in Industrial Engineering from the University of Illinois at Chicago in 2010. He is a Ph.D. candidate in the Department of Mechanical and Industrial Engineering. His research interests include digital signal processing, machinery health monitoring, bearing and gear fault diagnostics based on the evaluation of vibration and acoustic emission signals, and condition based maintenance.

Modeling Degradation Using Thermodynamic Entropy

Michael D. Bryant [1]

[1,] *Mechanical Engineering, University of Texas at Austin, Austin, Texas 78712-0292*
bryantmd@austin.utexas.edu

ABSTRACT

Manufacture transforms raw materials into finished components. Ageing and degradation of components, driven by dissipative processes, irreversibly alter material structures. The second and third laws of thermodynamics assert that these dissipative processes must generate entropy. This entropy is a fundamental quantity to describe ageing and degradation.

This recognition led to a Thermodynamic Degradation Paradigm encapsulated in a Degradation Entropy Generation (DEG) Theorem, wherein the rate of degradation was related to the irreversible entropies produced by the underlying dissipative physical processes that age and degrade components. This paradigm and theorem permit a structured approach to modeling degradation of any kind. If properly applied, the DEG Theorem leads to a differential equation in a variable that describes the degradation. The equation depends on the operational and environmental variables that characterize the system. Integration of the equation accumulates the degradation over time. This approach has led to accurate models for progression of and failure by wear, fatigue, and battery degradation that are consistent with prior models.

This article will review the Thermodynamic Degradation Paradigm and Degradation Entropy Generation Theorem, and apply these to formulate predictive models of wear, fatigue, and battery degradation, i.e., differential equations that govern the degradation or ageing. The article will conclude with a discussion on how to use these governing degradation equations for machine prognosis.

1. THERMODYNAMIC DEGRADATION PARADIGM AND DEGRADATION ENTROPY GENERATION THEOREM

Doelling, Ling, Bryant, and Heilman (2000) originally proposed the Thermodynamic Degradation Paradigm (TDP), which states that the irreversible entropy produced as a consequence of degradation can become a fundamental variable to quantitatively describe the degradation. For boundary lubricated sliding of copper on steel, they measured wear and the concomitant entropy produced at the sliding interface, and showed wear proportional to entropy. For dry sliding of bronze on stainless steel, Bryant and Khonsari (2008) also showed wear proportional to entropy.

Degradation manifests via a mechanism of dissipative processes. Dissipative processes that damage tribology interfaces include adhesion, surface plastic deformation, fracture, chemical reaction, material phase changes, viscous dissipation, heat dissipation, and material mixing, among others (Bryant, 2009).

Bryant, Khonsari, and Ling (2008) encapsulated the Thermodynamic Degradation Paradigm into the Degradation Entropy Generation (DEG) Theorem. Suppose degradation of whatever form can be measured by a variable w, which is a non negative, monotonic function $w=w\{p_i\}$ of the energies p_i associated with the $i = 1,2,\ldots,n$ dissipative processes that comprise the degradation mechanism. Suppose also that the processes $p_i = p_i(\xi^j_i)$ depend on time dependent phenomenological variables $\xi^j_i = \xi^j_i(t)$. Here i indexes all the dissipative processes of a degradation mechanism, and j indexes the phenomenological variables within a dissipative process. Altogether, $w=w\{p_i(\xi^j_i)\}$. According to thermodynamics laws 2 and 3, the dissipative processes *must* produce irreversible entropy $S' = S'\{p_i(\xi^j_i)\}$ dependent on the same energies and phenomenological variables. Via the chain rule, rates of entropy and degradation are (Bryant et al, 2008)

$$\frac{dS'}{dt} = \sum_{i,j}\left(\frac{\partial S'}{\partial p_i}\frac{\partial p_i}{\partial \xi^j_i}\right)\frac{\partial \xi^j_i}{\partial t} = \sum_i \frac{dS'_i}{dt} \qquad (1)$$

$$\frac{dw}{dt} = \sum_{i,j}\left(\frac{\partial w}{\partial p_i}\frac{\partial p_i}{\partial \xi^j_i}\right)\frac{\partial \xi^j_i}{\partial t} = \sum_{i,j}\left[\frac{\partial w/\partial p_i}{\partial S'/\partial p_i}\right]\left(\frac{\partial S'}{\partial p_i}\frac{\partial p_i}{\partial \xi^j_i}\right)\frac{\partial \xi^j_i}{\partial t}$$

Michael Bryant. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

$$= \sum_i B_i \frac{dS'_i}{dt} \tag{2}$$

In the foregoing, indices *i,j* beneath summation signs refer to a sum over both variables. The similar dependence structures in $w=w\{p_i(\xi^j_i)\}$ and $S' = S'\{p_i(\xi^j_i)\}$ led to the similar factors inside the sums in Eqs. (1) and (2). Terms in the first line of Eq. (2) were multiplied by unity in the form $(\partial S'/\partial p_i)^{-1}(\partial S'/\partial p_i)$ to give the terms in the second line of Eq. (2). Coefficients

$$B_i = \frac{\partial w}{\partial p_i}\left(\frac{\partial S'}{\partial p_i}\right)^{-1} = \left.\frac{\partial w}{\partial S'}\right|_{p_i} \tag{3}$$

which arise from the terms inside the square bracket of Eq. (2) are material properties that represent the increment of degradation incurred per increment of entropy generated by activity of process p_i. The B_i can be measured or related to other material properties. If p_i is the energy dissipated by a dissipative process, definition of entropy suggests $\partial S'/\partial p_i = 1/T_i$, where T_i is a temperature associated with p_i. The other terms in the second line of Eq. (2) are then dissipated power components $dp_i/dt = (\partial p_i/\partial \xi^j_i)(\partial \xi^j_i/\partial t)$.

The final line of Eq. (2) asserts that rate of degradation can be expressed as a linear combination of the rates of production of irreversible entropy by the underlying dissipative processes of a degradation mechanism. As suggested in the prior paragraph, when a process dissipates power dp_i/dt the irreversible entropy S' generated and the degradation w depend on process temperature T_i, generalized force $(\partial p_i/\partial \xi^j_i)$ and generalized velocity $(\partial \xi^j_i/\partial t)$.

Also used will be laws 1 and 2 of thermodynamics (Kondepudi & Prigogine, 1998)

$$dE = dQ - dW + \sum \eta_k dN_k \tag{4a}$$

$$dS = dS' + dS_e \tag{4b}$$

that conserve energy (internal energy E, heat Q, work W and energy in chemical potential η_k and molar mass N_k) and entropy (entropy flow S_e associated with heat flow, entropy generated S', and system entropy S).

2. APPLICATIONS

The DEG theorem presented in the prior section will be applied to sliding wear, fatigue, and battery degradation.

2.1. Sliding Wear

Figure 1 depicts a slider sliding on a counter surface. Bodies that rub and slide shed particles that accumulate as wear, measured as volume w of material lost. Friction force $F = \mu N$ sliding through distance x dissipates work $p = Fx$, where μ is friction coefficient and N normal load. For steady

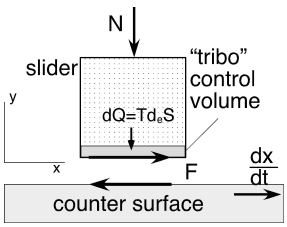


Figure 1. Slider on counter surface.

sliding at speed dx/dt the process is stationary, rendering $dE = dS = 0$ which simplifies Eqs. (4). Also, since the internal energy lost with a wear particle is small, the final term in Eq. (4a) can be neglected compared to other terms. The entropy produced arises from the friction work p dissipated within the tribo control volume (Fig. 1) that encompasses the sliding interface and nearby surface layers within slider and counter surface. Via Eq. (1), friction generates entropy at rate

$$\frac{dS'}{dt} = \left(\frac{dS'}{dp}\frac{dp}{dx}\right)\frac{dx}{dt} = \frac{F}{T}\frac{dx}{dt}, \tag{5a}$$

where contact temperature T arose via $dS'/dp = 1/T$. Applying Eq. (2),

$$\frac{dw}{dt} = \frac{BF}{T}\frac{dx}{dt} = \frac{B\mu}{T}N\frac{dx}{dt}. \tag{5b}$$

Archard's (1953) wear law $w = kNx/H$ relates w to N and x, via wear constant k and hardness H of the softer of the material pair. With N and T constant, Archard's law gives

$$\frac{dw}{dt} = \frac{k}{H}N\frac{dx}{dt}. \tag{5c}$$

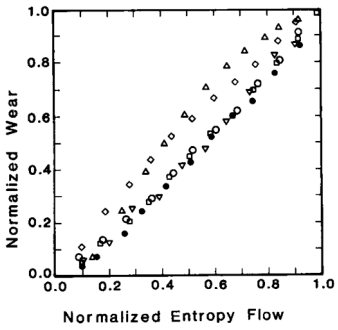


Figure 2. Wear vs. entropy flow, with axes normalized by max values. Symbols show six trials. Load N = 9.9 kg, speed dx/dt = 3.3 ms^{-1}. From Doelling et al (2000).

Comparing (5c) to (5b),

$$k = \frac{B\mu H}{T}. \tag{5d}$$

Doelling et al (2000) estimated $B = dw/dS'|_p$ under boundary lubricated sliding of copper on steel using the Fig. 2 graph of normalized wear versus normalized entropy flow.

Doelling calculated entropy flow $S_n = \sum^{n} \frac{\Delta Q^{(n)}}{T^{(n)}}$, where $\Delta Q^{(n)}$ is the heat input to the slider during the nth time interval, and $T^{(n)}$ is the corresponding average absolute surface temperature of the stationary copper slider rubbing the rotating steel cylinder. Via Eq. (4b) and a stationary process wherein $dS = 0$, $dS_e = - dS'$, leading to $B = dw/dS_e$. Averages over the six data trials of Fig. 2 led to $B = 4.0 \times 10^{-10}$ m^3/(JK^{-1}) and k=1.01×10^{-4}. For the same metals sliding under poor or similar lubrication, Rabinowicz (1980) measured k=1.0×10^{-4}. The k estimated by (5d) and Fig. 2 arose from measured wear, temperatures and forces; Rabinowicz's k via Archard (1953) arose from measured wear, forces and distance.

A different sliding configuration (bronze ring against 4140 stainless steel) under *dry rubbing* with different load 226.8 kg (500 lb), speed, temperature, and materials was performed (Bryant & Khonsari, 2008). Friction coefficient, wear, and temperature were measured similar to Doelling et al (2000). Results also yielded a remarkably accurate wear coefficient compared to Rabinowicz (1980).

Measurements of very different quantities rendering k values identical to within a few percent suggests validity of the entropy/degradation hypothesis, and Eq. (2). Note that integration of Eq. (5b) would accumulate the degradation by wear. Since energetics of degradation and friction are embedded in the entropy-wear statement of (5b), friction and wear are treated as related, not separate. Archard's wear law is a subset of the thermodynamic entropy formulation. As shown in Bryant et al (2008), Eqs. (2) and (3) can describe other modes of wear (abrasive and fretting among others) if the entropy generated can be formulated.

2.2. Metal Fatigue

Amiri & Khonsari (2012), Naderi & Khonsari (2010), and Amiri, Naderi & Khonsari (2011) have shown that fatigue of metals and other materials such as composites (Naderi & Khonsari, 2012) obey the Thermodynamic Degradation Paradigm and the Degradation Entropy Generation (DEG) Theorem. Collectively, these references experimentally correlate the cumulative effects of fatigue damage with the entropy produced in a fatiguing member.

Fatigue is driven by the energy dissipated by plasticity and fracture. Amiri et al (2011) formulated an expression for the irreversible entropy S' generated during progression of fatigue based on the Clausius-Duhem inequality (Lemaitre & Chaboche, 1990). But since the expression was very complicated——the expression included the complex plastic strain and stress fields, the thermal heat transfer, and the strain hardening effects——they instead pursued entropy flow S_e as a substitute for S' under a stationary process approximation $dS = 0$. As discussed in section 2.1, this leads to $dS_e = - dS'$, which allows use of entropy flow in the DEG theorem, to describe degradation. Naderi, Amiri, & Khonsari (2010) determined the entropy flow via temperature measurements over a cantilevered member undergoing reverse bending, using an infra-red camera in conjunction with thermal finite elements. Finite elements calculations of stress and heat transfer induced by plastic work estimated flows of heat and entropy over the fatigued member. The elasto-plastic-thermal finite element model was excited by mechanical loads similar to those applied to the specimen, and 2709 ten-node quadratic tetrahedral finite elements connected via an appropriate mesh. The temperatures estimated were consistent with the infra-red measured temperature distribution.

Naderi et al (2010) found that during an initial phase of the first hundred cycles or so temperatures increased due to heat generated by plastic work dissipation. During a second phase of thousand of cycles, temperatures stabilized at approximately constant levels set by equilibrium between heat transfer and heat generated. Finally, near the end of the component's fatigue life, temperatures abruptly increased. Naderi et al (2010) proposed that temperature can be used to predict progression of fatigue and fatigue failure.

Defining degradation measure w as rupture strength S_R (i.e., the maximum load the specimen can sustain) the DEG theorem gives $dS_R/dt = -BS'$ where the minus sign denotes diminishing. When integrated, Eq. (2) becomes

$$S_R = S_{R0} - B\int S' dt \,, \tag{6a}$$

where subscript 0 refers to the initial rupture strength. When sufficient entropy accumulates, fatigue strength S_R equals the applied load, and the specimen ruptures.

Similar to Fig. 2, Amiri, Naderi, & Khonsari (2011) found that a plot of normalized number of cycles M/M_f vs. normalized entropy flow S_e/S_f yielded an approximately linear function, up to the catastrophic rupture of the component. Their plot can be visualized by replacing normalized wear on the ordinate axis of Fig. 2 with normalized number of cycles. Amiri et al (2011) normalized S_e and M with counterparts S_f and M_f at rupture, since these were maximum values. Since the plot had unity slope, similar to Fig. 2, Amiri et al (2011) concluded

$$\frac{S_e}{S_f} \approx \frac{M}{M_f} \tag{6b}$$

which suggests that once a fatiguing member generates a critical amount of entropy S_f (which Eq. (6) correlates to a critical number of cycles M_f), rupture occurs. This is consistent to Eq. (6a) accumulating enough entropy to reduce the strength to critical levels. Indeed, others tests (Amiri et al, 2011) under different loading conditions such

as torsion showed the persistence of the same amount of critical entropy to rupture. Finally, Naderi & Khonsari (2010) showed that the damage parameter used extensively to characterize fatigue can be obtained from entropy.

2.3. Battery Degradation

In this section, a battery ageing model will be constructed from a model of the operational dynamics of a battery, by blindly applying the DEG theorem of Eq. (2) to those elements in the batteries' operational system dynamics model that generate entropy. The ageing model will be shown to be qualitatively consistent with known characteristics and traits of battery ageing.

Batteries store energy electrochemically. Popular battery types include lead acid and lithium ion batteries. Batteries consist of anode and cathode electrodes, electrolyte, separator, and terminals. Batteries have finite lifetimes, which are usually limited by manufacturing defects and ageing effects. This section will focus on ageing effects on battery life. Battery health is often measured in terms of capacity $\mathscr{C}$ (Ah), the amount of charge in ampere-hours a battery can deliver when discharged at a rated current, or growth of internal cell impedance Z (ohms), see (Broussely, Biensan, Bonhomme, Blanchard, Herreyre, Nechev & Staniewicz, 2005). Ageing reduces capacity $\mathscr{C}$ and increases impedance Z. Cycles of charging and discharging age a battery. Battery cycle life is rated as the number of complete charge-discharge cycles a battery can perform before 1) capacity $\mathscr{C}$ falls below 80% of initial rated capacity and/or 2) the internal resistance Z increases 1.3 to 2 times initial value.

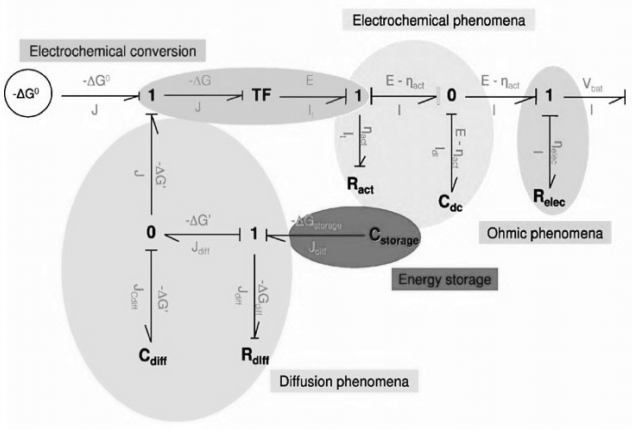


Figure 3. Bond graph systems dynamic model of a Li ion battery, from Ménard et al (2010).

Battery life, typically 500 to 1200 charge discharge cycles, depends on many factors (Broussely et al, 2005 and Vetter, Novak, Wagner, Veit, Moller, Besenhard, Winter, Wohlfahrt-Mehrens, Vogler & Hammouche, 2005). Most prevalent are:

(a) Number of charge-discharge cycles experienced by the battery: more cycles diminish remaining life.
(b) Depth of discharge (DOD- the percent of battery capacity discharged during a charge-discharge cycle): a larger DOD reduces cycle and increases the increment of energy dissipated per cycle.
(c) Electrolyte decomposition enhanced by high temperature and Li plating.
(d) Electrode plating by Li, which increases resistance and fades capacity, is exacerbated by lower temperature.

A battery operational model in bond graph form from Ménard, Fontès, & Astier (2010) models the dynamic electrochemical phenomena in a Li-ion battery. The bond graph of Ménard et al (2010) was copied and is presented as Fig. 3. A bond graph is a map of where and how power flows in a physical system. A bond graph also shows where energy is stored, dissipated, and transformed. The half arrows in Fig. 3 indicate the direction of positive power flow in the Li ion battery. In a bond graph, potential energy is stored in capacitance elements C, kinetic energy is stored in inertance elements I, and power is dissipated in resistance elements R. From a completed bond graph, the differential equations that govern the physics and dynamics of a system can be extracted (Brown, 2006 and Karnopp, Margolis, & Rosenberg, 2000).

The bond graph of Fig. 3 has additional labels to indicate where in the battery system is "energy storage", "diffusion phenomena", "electrochemical conversion", "electrochemical phenomena", and "ohmic phenomena". On the far right of Fig. 3 are battery terminal voltage V_{bat} and current I, which pertain to the voltage and current across the physical terminals of a physical battery. Chemical capacitance $C_{storage}$ found in the "energy storage" sector stores the battery's energy via electrochemical charge separation involving Li^+ ions and electrons. Gibbs free energy $-\Delta G_{\text{storage}}$ (J) and molar flow of lithium ions J (mol/sec) appear as effort and flow on multiple bonds, indicating the chemical thermodynamics embedded in this bond graph. The minus sign on $-\Delta G_{\text{storage}}$ refers to energy leaving the main storage $C_{storage}$. In "diffusion phenomena", capacitance C_{diff} and dissipative resistance R_{diff} together set a time constant which controls the slow dynamics of Li^+ ion diffusion in the electrolyte, which transports charge through the electrolyte. Transformer TF: nF in electrochemical conversion has modulus of the Faraday constant F (9.649 x 10^4 C mol^{-1}) and the number of moles of electrons n exchanged for each mole of lithium ions involved in the electrochemical reaction at the electrodes. This transformer, which converts electrochemical power to electrical power, bridges the electrochemical and electrical domains. Effort source $-\Delta G^0$ establishes a reference thermodynamic potential

within the bond graph. The power needed to activate electrochemical reactions at the electrode–electrolyte interfaces imposes resistance R_{act}. This activation power is not stored, but dissipated. Capacitance C_{dc} is due to a layer of charge (electrons and lithium ions Li^+) that forms about the electrode-electrolyte interface. Low mobility of Li^+ ions through the electrolyte relative to electron flows in the battery causes ohmic resistance R_{elec}.

In electric circuits, electric resistances dissipate power $V\,I$, where V and I are the voltage drop across and current through the physical resistor. The electric resistance R_{elec} in Fig. 3—which represents the Thevenin equivalent impedance seen across the battery terminal—has voltage drop η_{elec} and current I. Similar to an electrical resistance, the power dissipated by other resistances is the product of the labeled quantities (which are equivalent to voltages and currents) on the half arrow bonds. In the bond graph of Fig. 3, generalized resistances R_{elec}, R_{diff}, and R_{act} dissipate powers $\eta_{elec}\,I$, $-\Delta G_{diff}\,J_{diff}$, and $\eta_{act}\,I_t$ respectively.

The entropy S' generated is the dissipated power divided by a temperature associated with the dissipative process. Thus, via Eq. (1),

$$\frac{dS'}{dt} = \frac{-\Delta G_{diff} J_{diff}}{T_{diff}} + \frac{\eta_{act} I_t}{T_{act}} + \frac{\eta_{elec} I}{T_{elec}} \,. \qquad (7a)$$

By way of Eq. (2), the battery degradation is

$$\frac{dw}{dt} = B_{diff} \frac{-\Delta G_{diff} J_{diff}}{T_{diff}} + B_{act} \frac{\eta_{act} I_t}{T_{act}} + B_{elec} \frac{\eta_{elec} I}{T_{elec}} \,, \qquad (7b)$$

where degradation measure w can be battery capacity $\mathscr{C}$ and/or internal impedance Z. Temperatures T_{diff}, T_{act}, and T_{elec} are associated with the diffusion, activation, and electric domains of the battery. Equation (7b) relates rate of capacity or impedance change (Erdinc, Vural & Uzunoglu, 2009) to power dissipated, and sums dissipative effects from Li-ion diffusion into/out of electrodes, the energy of activation of Li/Li-ions at electrodes, and ohmic effects associated with mobility of Li ions in electrolyte. The sum over effects in Eq. (7b) is consistent with Vetter et al (2005), who reviewed ageing mechanisms in Li-ion batteries and stated that diverse "effects can be considered as additive". Equation (7b) is consistent with item (b) of the list, since a larger depth of discharge yields larger power dissipations from all effects, with greater per cycle changes in w, and with Broussely et al (2005) who found capacity faded and impedance increased with more charge-discharge cycles[1]. In Eq. (7b), coefficients B_{diff}, B_{act}, and B_{elec} should be adjusted to reflect the relative importance of each entropy term on the degradation. For w being $\mathscr{C}$ or Z, these coefficients must be negative or positive, respectively, to model capacity fade or impedance increase. Equation (7b) can be posed in terms of phenomenological variables via constitutive relations of Eqs. (14) and (17) of Ménard et al. (2010), wherein

$$-\Delta G_{diff} J_{diff} = \frac{RT_{diff}}{\alpha F I_{\lim}} I_t \,, \qquad \eta_{act} I_t =, \qquad \eta_{elec} I = R_{elec} I^2 \qquad (8a)$$

giving

$$\frac{dw}{dt} = B_{diff} \frac{RT_{diff}}{\alpha F I_{\lim}} \frac{I_t}{T_{diff}} + B_{act} \frac{RT_{act}}{\alpha F I_o} \frac{I_t^2}{T_{act}} + B_{elec} \frac{R_{elec} I^2}{T_{elec}} \,. \qquad (8b)$$

Here R is the molar gas constant, $\alpha = 1/2$, and I_{lim} and I_o are diffusion currents dependent on number of lithium ions. The first two terms of Eqs. (8b)—pertaining to electrolyte diffusion and electrode-electrolyte interface activation—increase influence of temperature on the electrolyte diffusion and activation terms of Eq. (7b), consistent with item list (c). The last term of Eq. (8b) retains T_{elec} in the electric term of Eq. (7b), suggesting more influence from this term at lower temperature, consistent with item list (d). Finally, since I_t increases with state of charge Ménard et al. (2010), a higher state of charge results in a higher, which increases the first two terms of Eqs. (7b) and (8b).

3. DISCUSSION

Prognostics tries to predict future behavior of systems, to assess performance. Models of a machine's physical operation, consisting of the machine system's governing differential equations, can accurately mimic machine behavior if the model has sufficient fidelity (i.e., the model includes the critical physics and has sufficient dynamic states), is given inputs similar to that of the real machine, and if the numerical values of the model's parameters accurately reflect the real machine's condition. Parameters can be tuned from data measured off the real machine. As equipment degrades or ages via processes such as wear, fatigue, and others such as those involved in battery ageing, the system's operational behavior can change, causing the machine to lose tolerance and not perform its function. Often this degraded machine behavior can be mimicked by the machine system's operational models (the governing differential equations), but with certain parameters associated with the ageing given revised numerical values. As a system ages, those parameters associated with the

[1] With each increment of energy dissipated during each charge-discharge cycle an increment of entropy must be produced, via the second law of thermodynamics. As cycles accumulate, the entropy produced accumulates, and list item (a) suggests battery life diminishes with increased entropy accumulation.

ageing will change with time. As presented in this article, physical ageing is driven by dissipative processes, and the ageing behavior can be predicted by solving differential equations posed in terms of a variable w that measures the degradation.

If degradation changes a certain parameter P_k then $P_k = P_k(w)$, and via the chain rule $dP_k/dt = dP_k/dw\ (dw/dt)$. Substitution of Eq. (2) then gives

$$\frac{dP_k}{dt} = \sum_{i=1}^{n}\left(B_i \frac{dP_k}{dw}\right)\frac{dS'_i}{dt} = \sum_{i=1}^{n} B_i^* \frac{dS'_i}{dt} \qquad (9)$$

where dP_k/dw was grouped with the "old" constants B_i to form new constants B_i*. Values for these constants can be derived by measurements of the ageing phenomena on a machine. Once tuned, governing differential equations can forecast the changes in parameters far into the future. These forecasted parameter values can then be used in the operational machine model to simulate future behavior of the aged or degraded machine.

4. CONCLUSION

The method presented derives differential equations that govern system degradation from an assessment of the irreversible entropy produced by operation of the system. Equation (2), which states the simple result of the DEG theorem, was applied to degradation venues of wear, fatigue, and battery ageing. When applied to wear, the DEG theorem and Eq. (2) related wear to friction and accurately described multiple forms of wear that do and do not obey Archard's wear law. When applied to fatigue, the DEG theorem and Eq. (2) accurately described fatigue, for reversed bending, torsion, and combinations, and led to a method of predicted fatigue life by measuring temperatures. Finally, when the DEG theorem and Eq. (2) were blindly applied to a system dynamics model of a battery, an expression for battery ageing Eq. (8b) was obtained that qualitatively agreed with existing observations of Li ion batteries.

ACKNOWLEDGEMENT

Research reported in this publication was supported in part by the National Science Foundation (NSF) grant IIP 1266279. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

NOMENCLATURE

B_i	degradation coefficient
$\mathscr{C}$	battery capacity (Ah)
C	generalized capacitance
E	internal energy
F	friction force
$-\Delta G$	Gibbs free energy for electrochemical cell
J	molar flow of lithium ions (mol/sec)
k	wear coefficient
M	number of fatigue cycles
N	normal force
p_i	energy of ith dissipative process
Q	heat
R	generalized resistance
S_e	entropy flow
S'	irreversible entropy generated
S	system entropy
S_f	critical entropy at failure
t	time
T	temperature
w	degradation
W	work
x	distance slid
Z	battery internal impedance
η	internal voltage drop in battery
μ	friction coefficient
ζ	phenomenological variable

REFERENCES

Amiri, M., Naderi, M. & Khonsari, M.M. (2011). An experimental approach to evaluate the critical damage, *International Journal of Damage Mechanics,* vol. 20 (89), doi: 10.1177/1056789509343082.

Amiri, M. & Khonsari, M. M. (2012). On the role of entropy generation in processes involving fatigue, *Entropy*, vol. 14, pp. 24-31, doi:10.3390/e14010024.

Archard, J.F. (1953). Contact and rubbing of flat surfaces, *Journal of Applied Physics*, vol. 24, pp. 981-988, doi:10.1063/1.1721448.

Brown, F.T. (2006). *Engineering System Dynamics, a Unified Graph-Centered Approach*, 2nd Ed., NY: CRC Press.

Broussely, M., Biensan, Ph., Bonhomme, F., Blanchard, Ph., Herreyre, S. , Nechev, K. & Staniewicz, K. (2005). Main aging mechanisms in Li ion batteries, *Journal of Power Sources*, vol. 146, pp. 90–96, doi:10.1016/j.jpowsour.2005.03.172.

Bryant, M.D. (2009). Unification of friction and wear, Chapter 4 of *Recent Developments in Wear Prevention, Friction, and Lubrication*, Research Signpost, Kerala, India, G. Nikas Ed., ISBN 978-81-308-0377-7, pp. 159-196.

Bryant, M.D. & Khonsari, M.M. (2008). Application of degradation-entropy generation theorem to dry sliding friction and wear," *Proceedings of STLE/ASME International Joint Tribology Conference*, Oct. 20-22, Miami, FL. IJTC2008-71079.

Bryant, M.D., Khonsari, M.M. & Ling, F.F. (2008). On the thermodynamics of degradation, *Proceedings of Royal*

Society of London Series A, vol. 464 (2096), pp. 2001-2014, doi:10.1098/rspa.2007.0371.

Doelling, K. L., Ling, F. F., Bryant, M. D. & Heilman B. P. (2000). An experimental study of the correlation between wear and entropy flow in machinery components, *Journal of Applied Physics,* vol. 88, pp. 2999- 3003.

Erdinc, O., Vural, B. & Uzunoglu, M. (2009). A dynamic lithium-ion battery model considering the effects of temperature and capacity fading, *2009 International Conference on Clean Electrical Power*, pp. 383-386, June 9-11, doi: 10.1109/ICCEP.2009.5212025.

Karnopp, D.C., Margolis, D.L. & Rosenberg, R.C. (2000). *System Dynamics, a Unified Approach,* 3rd ed., New York: Wiley.

Kondepudi, D. & Prigogine, I. (1998). *Modern thermodynamics from heat engines to dissipative structures*, New York, NY: John Wiley & Sons, Inc.

Lemaitre, J. & Chaboche, J.L. (1990) *Mechanics of solid materials 1st ed.*, Cambridge, UK: University Press.

Ménard, L., Fontès, G. & Astier, S. (2010). Dynamic energy model of a lithium-ion battery, *Mathematics and Computers in Simulation*. vol. 81, pp. 327–339, doi:10.1016/j.matcom.2010.07.026.

Naderi, M., Amiri, M. & Khonsari, M.M. (2010). On the thermodynamic entropy of fatigue fracture, *Proceedings of Royal Society of London Series A*, vol. 466, pp. 423–438, 10.1098/rspa.2009.0348.

Naderi, M. & Khonsari, M.M. (2010). A thermodynamic approach to fatigue damage accumulation under variable loading, *Materials Science & Engineering A*, vol. 527, pp. 6133–6139, doi:10.1016/j.msea.2010.05.018.

Naderi, M. & Khonsari, M.M. (2012). A comprehensive fatigue failure criterion based on thermodynamic approach, *Journal of Composite Materials,* vol. 46 (437) doi: 10.1177/0021998311419540.

Rabinowicz, E. (1980). *Wear control handbook*, New York, NY: ASME press, pp. 486.

Vetter, J. , Novak, P. , Wagner, M.R. , Veit, C.,. Moller, K.C., Besenhard, J.O., Winter, M., Wohlfahrt-Mehrens, M., Vogler, C. & Hammouche, A. (2005). Ageing mechanisms in lithium-ion batteries, *Journal of Power Sources*, vol. 147, pp. 269–281, doi:10.1016/j.jpowsour.2005.01.006.

BIOGRAPHY

Michael D. Bryant, Accenture Endowed Professor of Mechanical Engineering at University of Texas at Austin, Austin TX, USA, was born in Danville IL on Feb. 8, 1951. His education includes B.S., Bioengineering, University of Illinois at Chicago, Chicago, IL, USA, 1972; graduate study in Information Engineering, University of Illinois at Chicago, 1972-1974; M.S., Mechanical Engineering, Northwestern University, Evanston, IL, USA, 1980; and Ph.D., Engineering Science and Applied Mathematics, Northwestern University, 1981. He has been with University of Texas at Austin since 1988. From 1981-1988 he was an Assistant and Associate Professor at North Carolina State University. His interests include tribology, mechatronics, manufacturing, and system modeling. He is a fellow of American Society of Mechanical Engineers, was Editor in Chief of ASME Journal of Tribology from 2005-2012, and is a member of Institute of Electrical and Electronics Engineers and Sigma Xi.

Electrochemical Impedance Spectroscopy for Coating Evaluation using a Micro Sensor

Nicholas Waters[1], Richard Connolly[2], Douglas Brown[3], and Bernard Laskowski[4]

[1,2,3,4] *Analatom Inc., Santa Clara, CA 95054, USA*

Nicholas.waters@analatom.com
Richard.Connolloy@analatom.com
doug.brown@analatom.com
Bernard.Laskowski@analatom.com

ABSTRACT

This paper discusses a micro-linear polarization resistance (μLPR) sensor modified to perform coating evaluation by means of electrochemical impedance spectroscopy (EIS). A circuit model is used with the EIS data to measure solution resistance, pore resistance, charge transfer resistance, intact coating capacitance, and double layer capacitance. These measurements allow the end user to monitor degradation of protective coatings in real-time, through non-destructive means. This is demonstrated through an accelerated aging test using a coated metal plate with a modified μLPR sensor. A metal panel made from aluminum alloy 7075-T6 was coated with 2 mils of an epoxy-based polymer coating and 2 mils of high solids polyurethane. The sensor was adhered to the face of the coated panel in a manner that allowed the electrolyte solution consisting of 3.5% NaCl to flow between the sensor and the coated surface of the panel. EIS measurements were acquired every hour for a total of 35 hours and at the conclusion of the test, changes in key parameters within the circuit model identified the initial time and mechanism of coating degradation, in this case, delamination.

1. BACKGROUND

Polymer coatings are commonly applied to metal substrates to prevent contact with natural elements that initiate and perpetuate corrosion. This corrosion process requires the metal be in contact with oxygen and an electrolyte. Protective coating integrity is of utmost importance to maximize remaining useful life of equipment and minimize costs associated with maintenance and repairs. Electrochemical impedance spectroscopy (EIS) provides a means of monitoring the present condition of a protective coating. Small defects in the protective coating, if undetected and unaddressed, can lead to coating failures, thus providing pathways for the electrolyte to reach the metal substrate.

Nicholas Waters et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The British Standards Institution's (BSI) Publicly Available Specification for the optimized management of physical assets defines asset management as the "systematic and coordinated activities and practices through which an organization optimally and sustainably manages its assets and asset systems, their associated performance, risks and expenditures over their life cycles for the purpose of achieving its organizational strategic plan." The motivation for effective asset management is driven by owners' desire for higher value assets at less overall costs, thus extracting the maximum value from their assets (Engineering, 2012). Condition-based maintenance aims to maximize asset value by extending the useful life of assets through mitigation of unnecessary maintenance actions performed during schedule based maintenance strategies. By providing maintenance engineers with information regarding the health of the structure, maintenance can be performed on a basis of necessity unique to each asset, as opposed to schedule-based predictions formed on statistical trends of operational reliability.

Protective coatings are the first line of defense against corrosion for metal substrates. Coatings aim to prolong the integrity of metal structures by creating a barrier between the elements and the metal substrate. Removing the possibility of contact with electrolytic fluid prevents electron transfer between the anodic and cathodic portions of the metal, which prevents the oxidation-reduction reactions that lead to metal loss. EIS measurements evaluate the integrity of the protective coating and are the first indication of compromised structural health of an asset.

The micro-linear polarization resistance (μLPR) corrosion sensor presented in this paper, provides insight into the health of coated metal structures through non-destructive

testing. In its native configuration, this sensor is capable of identifying coating failure through changes in polarization resistance and time-of-wetness measurements which is further explained in (Brown, 2014). By measuring changes in the electrochemical properties of the coating, EIS is able to monitor coating degradation over time. Coupling EIS with linear polarization resistance provides a broader assessment of structural and coating integrity by answering "why" and "how" failure occurs on susceptible components.

EIS can be used as a non-destructive method of performing coating evaluation in real time. Impedance values of the electrochemical cell are determined by applying a sinusoidal voltage at various fixed frequencies and measuring the current response. Impedance is calculated from the current's magnitude and phase response with respect to the applied potential across an electrochemical cell. Typically, EIS measurements are represented by either Bode or Nyquist plots. After acquiring EIS data, a circuit model representing the impedance of the coating is selected that provides the best fit for experimental data. Once the appropriate model is selected, it is possible to extract values for model parameters, such as resistance and capacitance. EIS provides insight into how each parameter changes by the electrochemical properties of the coating as the coating degrades over time; this provides insight into the level and type of degradation taking place (David Loveday, 2004; Gamry Instruments, 2011).

2. LITERATURE REVIEW

Coating degradation is a costly problem that many industries face. The best way to minimize costs associated with corrosion is to mitigate the effects through preventative conservation. Similar to the metal substrate, the coating degrades over time leaving the metal exposed to the elements. Providing service engineers insight into the state of their protective coatings is not only critical when dealing with valuable equipment, but also in the preservation of historical artifacts, where corrosion is taking place on priceless historical pieces (Emilio Cano, 2010).

Proper application of the coating is one of the main factors affecting lifetime and performance. Improper application of the coating can lead to poor adhesion to the metal substrate which provides pathways for corrosive substances to undercut the coating and compromise the coating's ability to protect the metal from corrosion. EIS provides a means of monitoring and evaluating the key parameters that change as the coating degrades over time, providing the user an opportunity to intercept the degradation pathways with preventative maintenance strategies (Api Popoola, 2014; M. Taqi Zahid Butt, n.d.).

3. IMPEDANCE AND EQUIVALENT CIRCUIT FOR COATING EVALUATION

The use of EIS to measure coating degradation relies on impedance measurements. Impedance is a measure of a circuit's ability to resist current and is defined as the ratio of the applied voltage to the current. A small amplitude sinusoidal excitation signal is applied across the coating. The amplitude of this excitation signal must be low, as the simple linear relationship relating resistance to current and voltage, shown in Eq. (1), becomes non-linear with more complex circuits.

$$R = \frac{E}{I} \text{ , (equation for an ideal resistor)} \quad (1)$$

where E is the voltage and I is the current. In more complex non-linear systems, impedance is the metric used to represent the circuit's ability to resist the flow of current. By applying a small amplitude excitation potential to the electro-chemical cell, it is possible to observe a pseudo-linear response in the response current which is shifted in phase. This excitation potential is expressed according to Eq. (2)

$$\vec{E}_t = E_o \sin(\omega t), \quad (2)$$

where $\vec{E_t}$ is the applied potential, E_o is the amplitude of the applied potential, and ω is the radial frequency (2πf). The response current is expressed according to Eq. (3)

$$\vec{I}_t = I_o \sin(\omega t + \phi), \quad (3)$$

where $\vec{I}_t$ is the response current, I_o is the amplitude of the response current, ω is the radial frequency, and ϕ is the phase. The impedance is then defined as the ratio of the applied potential to the response current as shown in Eq. (4)

$$Z = \frac{E_t}{I_t} \ . \quad (4)$$

A potentiostat is used to apply a frequency sweep of the potential across the electrochemical cell and measure the response current. These data are then used to calculate the resulting impedance. Data is plotted using a Bode plot which displays phase and impedance as a function of frequency. EIS relies on fitting a model to impedance values based on an equivalent circuit representation of the interrogated electrochemical system. Impedance values for different circuit components are listed below in Table 1.

Table 1. Circuit components and corresponding impedance values.

Circuit Component	Impedance (Z)
Resistor	R
Inductor	$J\omega L$
Capacitor	$1/(J\omega C)$

Where R = Resistance, ω is radial frequency, L is inductance, $J = \sqrt{-1}$, and C is capacitance.

For a linear system and circuit components wired in series (Figure **1**), the equivalent impedance value is calculated according to Eq. (5).

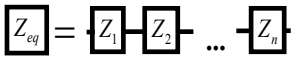


Figure 1. Circuit components wired in series.

$$Z_{eq} = Z_1 + Z_2 + \ldots + Z_n \ . \tag{5}$$

For circuit components wired in parallel (Figure **2**), the equivalent impedance value is calculated according to Eq. (6).

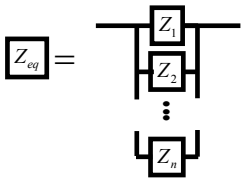


Figure 2. Circuit components wired in parallel.

$$\frac{1}{Z_{eq}} = \frac{1}{Z_1} + \frac{1}{Z_2} + \ldots + \frac{1}{Z_n} . \tag{6}$$

In order to use EIS to perform coating evaluation, a circuit model is used to represent the physical system comprising the electrochemical cell. A coated metal plate is wired as the working electrode and is submerged in an electrolyte. Reference and counter electrodes are placed in the electrolyte as well. As an alternating potential is applied to the working electrode (the coated panel), the metal substrate, coating, and electrolyte form a capacitor, whose value is referred to as the coating capacitance (Cc). The metal substrate and electrolyte form parallel plates, while the coating acts as the dielectric barrier. An additional capacitor is formed when the coating begins to delaminate and electrolyte has penetrated the space between the coating and the metal substrate. The electrolyte and the metal form the two plates of the capacitor, while a single layer of water molecules (Helmholtz Plane) separates the two plates forming the dielectric. This capacitance is referred to as the double layer capacitance (C_{dl}). The circuit model shown in Figure 3 is commonly used to represent metal with protective coatings (Loveday, 2004; Mike O'Donoghue, 2003).

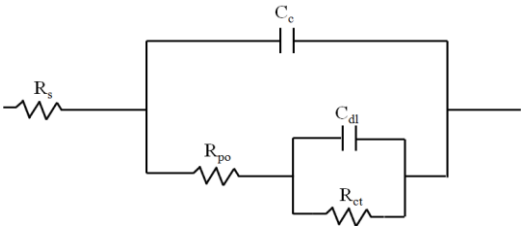


Figure 3. Equivalent circuit diagram for paint used for EIS.

In the circuit model, R_s is the solution resistance, R_{po} is the pore resistance, C_c is the intact coating capacitance, C_{dl} is the double layer capacitance, and R_{ct} is the charge transfer resistance. Once the model has been fitted to the data, changes in the model's parameters offer insight into the health of the coating. For example a decrease in coating capacitance represents deterioration of the coating's ability to shield the metal substrate from the environment. Another example is the pore resistance, which provides information on the effectiveness of the coating. As pores in the paint begin to expand over time, the resistance associated with these pores decreases. This parameter provides a general indication of paint degradation (Gamry Instruments, 2011; K. M. Deen, 2009). Figure **4** provides a physical representation of the circuit model used to interpolate the impedance data.

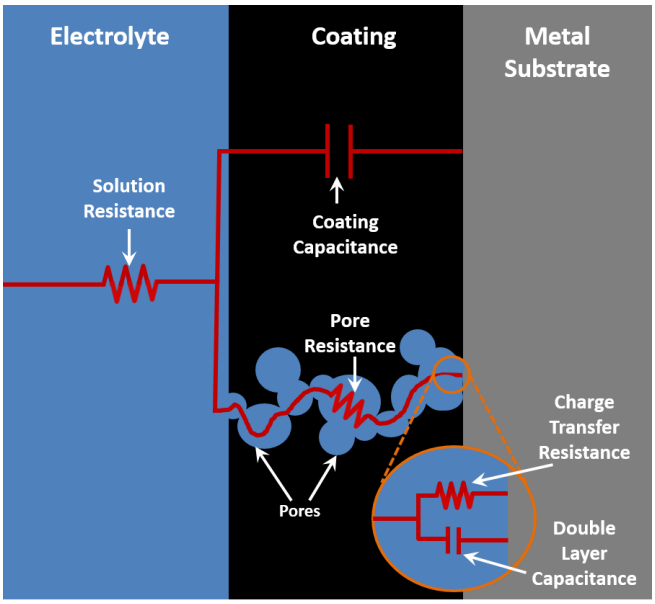
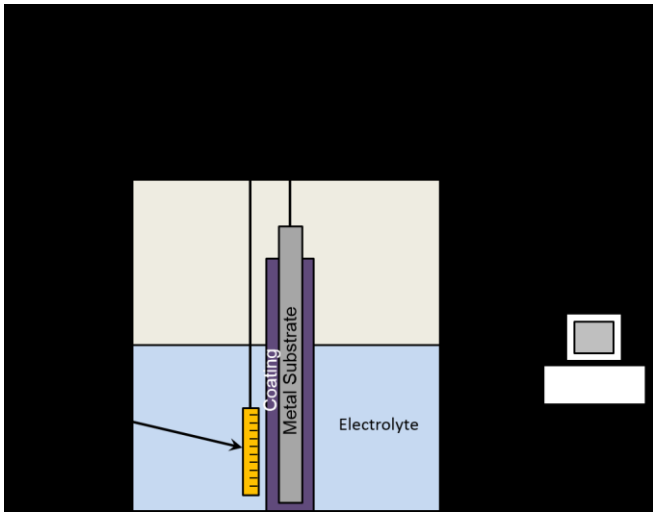


Figure 4. Physical representation of the equivalent circuit model for damaged coating.

4. EXPERIMENTAL RESULTS

4.1. Test Plan

Research is currently being conducted using a modified µLPR corrosion sensor for EIS measurements. The first EIS test cell is configured as depicted below in Figrue **5**.



Figrue 5. EIS experimental setup depicting a coated metal panel acting as the working electrode and a two electrode sensor connected to the counter and reference electrodes.

First, a metal panel made from aluminum alloy 7075-T6 is coated with 2 mils of an epoxy-based polymer coating and 2 mils of high solids polyurethane. The sensor is then adhered to the face of the panel with industrial strength epoxy. The bonding agent (industrial strength epoxy) is placed on opposing edges of the sensor so as to adhere the sensor to the surface of the painted metal plate in a manner such that the ambient environment is allowed to rapidly diffuse between the sensor and the painted substrate. The coated metal plate is then connected to a potentiostat as the working electrode. Two leads are connected to the sensor; one as the counter electrode and another as the reference electrode. A baseline EIS measurement is then taken with the sensor and panel in ambient air. The coated panel/sensor configuration is then placed in a solution containing 3.5% sodium chloride in deionized water. Another EIS measurement is taken immediately after submerging the panel/sensor configuration. EIS measurements are then taken every hour following submersion in the electrolyte solution. A circuit model is then selected based on the fit criteria between the expected and acquired EIS data.

Figure 6. Singleton corrosion test chamber used to run ASTM G85 A5 cyclic fog test.

Coating evaluation is also currently being conducted using the modified µLPR for coated panels in a Singleton corrosion test chamber, shown in Figure **6**. A panel coated with 4 mils of an epoxy-based polymer coating and 2 mils of high solids polyurethane was placed in a Singleton corrosion test chamber. Prohesion testing is being performed following the ASTM G85 Annex A5 Dilute Electrolyte Cyclic Fog/Dry Test. This test consists of a 1 hour fog at 25°C followed by a 1 hour dry-off period at 35°C. The electrolyte used for the fog is made up of 0.05% sodium chloride and 0.35% ammonium sulfate by mass in deionized water.

4.2. Results

To test the system's ability to perform coating evaluation in a typical laboratory environment, an experiment was conducted. A metal panel made from AA 7075-T6 measuring 7.6 cm x 1.91 cm x 0.16 cm was used for this accelerated coating evaluation experiment. Three quarters of the panel was coated with 2 mils of an epoxy-based polymer coating. A µLPR sensor was adhered to the face of the painted portion of the panel. The working lead of the potentiostat was connected to the uncoated portion of the panel. The counter electrode and reference were connected to the µLPR sensor as shown in Figure **7**. A 10 mV AC signal operating between 10 mHz and 10 MHz was utilized as the interrogation waveform. The coated panel was partially submerged in a graduated cylinder containing 3.5% sodium chloride such that only the coated portion of the plate was submerged while the uncoated portion of the plate and working electrode interface were outside the solution.

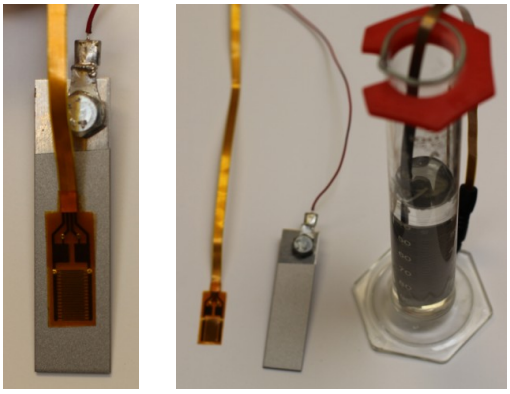
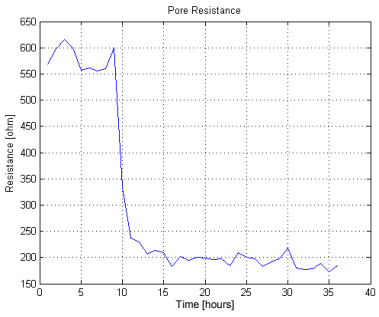
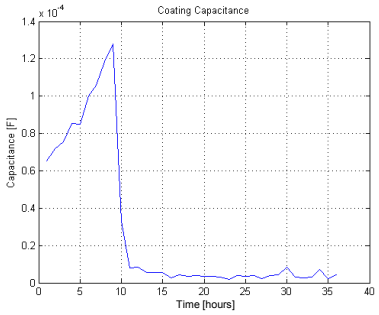


Figure 7. Photo of sensor and coated panel configuration (left) and sensor, coated panel with working electrode, and graduated cylinder with panel partially submerged in solution of 3.5% sodium chloride (right).

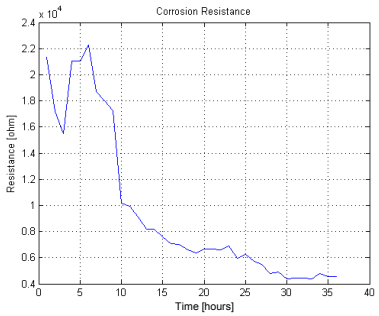
Data collection was set at one-hour intervals. The plots shown display the changes in R_{po}, C_c, R_{ct}, and C_{dl} over the 35 hours of data collection for the submerged panel. Once the panel is placed in the solution, the coating begins to absorb electrolyte through its pores. This process causes the coating thickness to expand. As the coating absorbs fluid, the dielectric constant for the coating increases, causing an increase in coating capacitance, which is observed in the first 8 data sets, as shown in Figure **8**. After around 9 hours, a drastic drop in R_{po}, C_c, and R_{ct} was observed, indicating electrolyte penetrated through to the metal substrate (coating failure). At the time of coating failure it was observed that there was an increase in C_{dl}. This increase in capacitance can be attributed to electrochemical reactions occurring on the surface of the metal. After removing panel from the solution, regions of paint delamination were present across both faces of the plate.



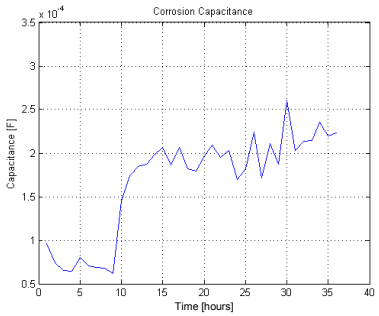
(a)



(b)



(c)



(d)

Figure 8. Plots of the pore resistance (a), coating capacitance (b), charge transfer resistance (c), and double layer capacitance (d) collected at 1 hour intervals.

5. CONCLUSION

In this paper, a μLPR sensor was used with EIS for coating evaluation. An accelerated corrosion test was performed on a coated metal plate. EIS data was collected over 35 hours which showed a sharp decrease in R_{po}, C_c, and R_{ct} and a sharp increase in C_{dl} during the duration of the experiment. The data showed failure of the protective coating 9 hours into the test, due to the thin coating layer and high salt concentration. Key parameters were evaluated within the circuit model to identify the mechanism of coating degradation. Further, this experiment showed the shielding present on Analatom's micro-sensor was sufficient to reduce the effects of ambient electromagnetic interference when operating outside of a Faraday cage.

6. FUTURE WORK

Future work is necessary to further understand the relationship between mechanisms of coating failure and key modeling parameters. This will involve operating under more stringent conditions, such as in a corrosion chamber running the ASTM B117 profile. Testing within a corrosion chamber presents challenges due to the additional electromagnetic interference generated by the chamber and the inability to enclose the electrochemical cell within a Faraday cage. Multiple coating types will need to be tested. Experiments involving coated metal samples with controlled coating defects need to be conducted to attain information with regard to the fault propagation rate as well as the radius of detection for the μLPR sensor.

REFERENCES

Engineering Asset Management: Issues and Challenges. University of Cambridge, Cambridge, UK, Executive Briefing, 2012 Service Week Workshop on the Future of Asset Management, 2012.

Brown, D. W., Connolly, R. J., Darr, D. R., & Laskwoski, B. C. (2014). *Linear Polarization Resistance Sensor Using the Structure as a Working Electrode* (pp. 1-7), July 8-10, Nantes, France.

Loveday, D., Peterson, P., and Rodgers, B. 2004. Evaluation of Organic Coatings with Electrochemical Impedance Spectroscopy. *JCT coatingstech*, 2(13), pp.22-27.

Cano, E., Lafuente D., and Bastidas, D. M. , 2010. Use of EIS for the Evaluation of the Protective Properties of Coatings for Metallic Cultural Heritage: a Review. *J. Solid State Electrochem*, 14(3), pp.381-91.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, vol. 22, pp. 679-688. doi:10.1016/j.ijforecast.2006.03.001.

Gamry Instruments, 2011. *Rapid Electrochemical Assessment of Paint*. Application Note. Warminster, PA: Gamry Instruments.

Deen, K. M. and Khan, I. H. 2009. Corrosion Protection Evaluation of Mild Steel Painted Surface by Electrochemical Impedance Spectroscopy. *Journal of Quality and Technology Management*, V1, paper 6.

Popoola, A., Olorunniwo, O. E., and Ige, O.O., 2014. Corrosion Resistance Through the Application of Anti-Corrosion Coatings.

Butt, M. T. Z., Deen, K.M., and Majeed, U. n.d. Evaluation of the Protective Performance of Epoxy Coated Mild Steel by Electrochemical Impedance Spectroscopy. Online. University of the Punjab.

O'Donoghue, M., Garrett, R., Datta, V., and Roberts, P., 2003. Electrochemical Impedance Spectroscopy: Testing Coatings for Rapid Immersion Service. *Materials Performace*, pp.36-41.

BIOGRAPHIES

Nicholas Waters, M.S. earned his bachelor's degree from the University of California, Davis in Applied Mathematics. In 2012, he earned his master's degree in Ocean Engineering from Florida Atlantic University under the advisement of Dr. Pierre Philippe Beaujean. His graduate field of study was focused on machine condition monitoring and development of a vibration condition monitoring system for a 20 kW ocean turbine. Currently, Nicholas works for Analatom, Inc. in Santa Clara, CA as a project engineer. His current responsibilities include writing and submitting proposals for grants, conducting corrosion experiments with Analatom's μLPR sensor, and researching the μLPR sensor's ability to be repurposed to perform EIS for coating evaluation.

Richard Connolly, Ph. D. is the Senior Research Engineer for Analatom. Dr. Connolly holds a bachelor of science degree in chemical engineering and a doctor of philosophy degree in biomedical engineering. He is a fellow of the National Science Foundation and is an expert in interfacing engineering devices with skin. He has extensive experience in electrochemistry, bioelectrics, gene therapy, and plasma physics. Much of this experience was gained developing novel electrical systems for the prevention and treatment of disease. During his tenure at Analatom he has overseen testing and validation of electrochemical structural health monitoring technologies for aerospace and civil engineering applications.

Douglas Brown, Ph. D. is the Senior Systems Engineer for Analatom. Dr. Brown has eight years of experience developing and maturing prognostics & health management (PHM) and fault-tolerant control systems in avionics applications. He received the B.S. degree in electrical engineering from the Rochester Institute of Technology in 2006 and the M.S./Ph.D. degrees in electrical engineering from the Georgia Institute of Technology in 2008 and 2011, respectively. He is a recipient of the National Defense Science and Engineering Graduate (NDSEG) Fellowship and has received several best-paper awards for his work in PHM and fault-tolerant control.

Bernard Laskowski, Ph. D. Obtained a Ph.D. in Physics from the University of Brussels, Brussels, Belgium. As president of Analatom Inc,.Bernard managed 85 university, government, and private industry contracts. He received the U.S. Small Business Administration Administrator's Award for Excellence. He licensed MEMS technologies and obtained patents for the Analatom MEMS projects. Responsible for having obtained $3.5 M in US Air Force, Army, and Navy government seed funding for SHM MEMS projects.

A Time-domain Modeling and Simulation Framework for Comparative Analysis of Prognostics, Reliability and Robustness in System Design

Nicholas A. Lambert[1], Kyle B. Ferrio[2], and Douglas L. Goodman[3]

[1, 2, 3]*Ridgetop Group, Inc., Tucson, AZ, 85741, USA*

nlambert@ridgetopgroup.com
kferrio@ridgetopgroup.com
dgoodman@ridgetopgroup.com

ABSTRACT

Redundancy is an effective, high-level solution to the requirement for reliable safety-critical systems, but it comes at the cost of Size, Weight and Power (SWaP) and reduced capability. A modeling and simulation framework was developed to address the need for robust design alternatives to redundancy. Robustness, in our application, is treated as the insensitivity of the performance with reference to specification. The necessity to characterize both reliability and robustness in the same framework has resulted in a time-domain simulation approach to modeling behaviors associated with unreliability and a lack of robustness. The incorporation of these features offers a novel insight into potential applications of prognostic technology. Further development of this approach has the potential to allow designers to choose how risks associated with failures are mitigated, by redundancy, robustness, or prognosis.

By modeling the life of parts, the factors that impact them and the resulting behaviors, the observability and predictability (even controllability in the case of optimized, fault-tolerant, closed-loop control) of faults and failures is identified. Designers can determine which parts of a system would benefit from prognostic health management (PHM) technologies, adaptive / tolerant features to yield robust design, or redundancy based approaches. The complex causality in the models requires a Monte Carlo approach analogous to the simulation of fleets of systems; this, combined with the ability to simulate systems made from new and old parts, can inform strategies for condition-based maintenance (CBM).

We present the mathematical modeling concept and the simulation framework which permits comparative assessment of reliability, robustness and prognostics. The multi-hierarchical, systems integration aspects inherent to the concept make this technique highly applicable to real-world dynamic systems. The framework also supports statistical, standards based and physics-of-failure descriptions of stress, aging, fault and failure behaviors in a unified way. There are challenges to be overcome in realizing the benefits of this approach to model-based system design. Issues of model validation, data availability and computational burden are recognized and discussed. As we show, these challenges can be overcome to produce new design tools providing better products and transparent project quality.

Nicholas A. Lambert et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

1.1. The Requirement for a Unified Modeling Approach

Complexity is the arch-nemesis of the systems engineer. This has been addressed in a historical context in work by Zio (2009), where the need to develop methods for integrating dynamics and reliability analysis was highlighted. Reliability engineering methods employ methods that combat complexity by reducing a system to a list of its parts or by offering abstracted representations in the form of reliability block diagrams and fault trees. These typically have a much greater degree of abstraction than the detailed models which describe the dynamic behavior of the system, where causal relationships are the topic of interest.

Mathematical descriptions of system behaviors often take the form of differential and algebraic equations (DAEs), and comparable representations exist for discrete time, state, space, and event systems. Numerical integration methods and solvers are used to produce simulated solutions to the mathematical system representations. The simulations are used for testing of designs with reduced or no physical hardware representation of the system. They often represent the physical plant for development and testing of software.

Models of the system dynamics are computationally expensive to run and simulations of timescales associated with reliability are infeasible. As a result, reliability

considerations are omitted save for functionality for fault injection. In large projects and organizations, these starkly different modeling modalities are often implemented by separate teams, each with separate requirements and tools. Each team's input into the design decision making process occurs at different stages in the design of systems and this can miss opportunities for whole-system optimization, potentially producing sub-optimal solutions. The impacts to a high-level, abstracted reliability analysis of low-level design decisions made using the detailed dynamic models can be poorly communicated, understood or missed entirely given the organizational and methodological disparities that are inherent in the design of large, complicated systems. This issue has been identified and addressed by Siu (1994) and discrete event, explicit state-transition and extended reliability methods were reviewed; the methods described here approach the issue from a starting point of simulation of system dynamics.

1.2. A Novel Reliability Modeling Method

The primary focus of this work has been to assess robustness. Robustness is usually treated as a beneficial insensitivity of a design to variations in conditions or design parameters (for example, variation within manufacturing tolerance of component values), where the performance against the specification is used in assessing robustness. In this case, however, the question of robustness is with regard to a particular instance of a system. Is a given instance of the system design robust? In answering this, it was necessary to produce models of systems that lacked robustness. These models needed to exhibit features of variation, aging, degradation and failure in response to simulated usage. The measure of robustness used is also closely related to reliability, but rather than reporting the statistics of failure, the statistics of specification violations are used.

The incorporation of these aspects of system behavior makes it possible to include prognostic technologies. Through the mathematical modeling of fault and failure behavior that is accurate in its stochastic and deterministic properties, the attention of the designer can be focused on that which is predictable and where appropriate investments on prognostics can be made.

A number of challenges remain and are associated with computational feasibility (in this case of sequential and parallel Monte Carlo simulations) and verification and validation of modeling assumptions. This work presents the opportunity to unify system design practices by introducing time-domain simulation techniques that also serve as reliability predictions; the ability to assess robustness and prognostics as risk mitigating design features means that this topic will be applicable in safety and capability critical systems.

The following sections outline the techniques used for modeling and simulating unreliable systems and including behaviors from standards, statistical and physics of failure based approaches.

2. TIME DOMAIN MODELING OF UNRELIABLE SYSTEMS

Time-domain modeling serves as a useful tool for system integration. The behaviors of parts can be defined and their roles in the system interpreted, yielding the performance of the system as a whole. The methods presented here are intended to be used in the same way. There are models for aging, fault and failure behaviors associated with the usage of each part within a system. Changes that occur in parts are then represented in the system performance. For this method to offer some utility, it must be used as a system integration tool. In describing a part, there is little to be learned about the part; but by including that part in a system, we can learn something about the impact of the part behaviors on the system. We can also derive knowledge about system behaviors on individual parts. It is this causal loop that is the subject of investigations using this approach. There are two key questions to be answered:

1. Does the reliability and life performance of one part affect the reliability and life performance of other parts in the system?
2. Can knowledge of this be used as the basis for predictions about the behavior of individual parts and systems?

The first of these question aims to address challenges present in the design of ever more complex systems. The second question is regarding whether an enhanced understanding of system reliability behavior can be used to formulate effective prognostic solutions. A feature of the approach is that it allows for multiple and various representations of unreliable parts and systems to exist in the same modeling framework.

2.1. The Life State Approach

The key to modeling unreliable systems in a manner which fits with numerical integration based simulation techniques is to use a method involving a life state. This life state is a measure of the age of a part of a system which is analogous to a measure of time; however, the rate at which life elapses is linked to usage via a stress factor. For each part, the life state is the underlying variable upon which all age related behaviors are dependent. This is based on the fundamental reliability relationship found in MIL-HDBK-217:

$$\lambda_P = \lambda_b \pi \quad (1)$$

The predicted rate of failure is the product of the base rate of failure and the part stress factor. The part stress factor is unitless, but by considering the same equation expressed in terms of mean time to failure (MTTF), it can be deemed that

the unit of the part stress factor is hours per hour. It is the ratio of the predicted failure time to the base failure time. The part stress factor is the rate at which a part accumulates age; it is the rate (with respect to time) associated with the life state. This method yields a measure of physical age (referred to as "life") and chronological age as measured in elapsed time.

For complex parts, a vector approach can be taken such that a single part can have an n-dimensional life state with each state having its own stress factor function and accumulation properties. This feature permits multiple behaviors to be modeled.

Typical application of the part stress factors method requires estimation of nominal or maximal usage characteristics and operating temperatures. The life state approach allows for usage characteristics to be taken from time domain simulation results and integrated numerically with respect to time. Consider the Arrhenius relationship at the heart of the part stress approach:

$$\frac{dM}{dt} = Ae^{-\frac{E_a}{kT}} \tag{2}$$

M is the state of a chemical reaction process. If we consider the temperature, *T*, to be a function of time, *T(t)*, then numerical integration can be used to simulate the progression of the state, *M*.

One method for estimating the reliability of a part is to take a time averaged rate of life with respect to time and use a first-order prediction of when the life would reach the base mean time to failure. A more representative method is to re-estimate the part stress distribution in a system as the accumulation of stress into life results in changes of the physical properties of each of the parts. The physical properties will be referred to subsequently as *part parameters*.

2.2. Failure and Fault Onset Distributions

The use of predicted and base rates of failure is indicative of the single parameter exponential failure distribution; however, many distributions can be used in the analysis of reliability and these are supported by the life state approach.

Probability distributions are used to describe the random failure behavior of a population of systems, products, or test articles. The occurrence of failure events is typically described as a probability density function (PDF), cumulative distribution function (CDF), or hazard rate, expressed as functions of time. The life state approach sees these expressed as a function of the life state, rather than time.

The use of Bernoulli trials using uniformly distributed random numbers and the hazard rate for each distribution allows for the occurrence of fault onset and failure events in keeping with the distribution. This can be performed online, using numerical integration methods to derive the hazard rate, or offline where a set of events are pre-computed as crisp thresholds for comparison to a life state.

$$PDF = f(x) \tag{3}$$

$$CDF = F(x) = \int_0^\infty f(x)\,dx \tag{4}$$

$$h(x) = \frac{f(x)}{1 - F(x)} \tag{5}$$

Note that for exponentially distributed events, the hazard function is constant and the memoryless property is preserved in spite of the inclusion of the life state.

Where fault onsets and failure events are causally linked (i.e. the fault leads to the failure), the failure event can be associated only with life accumulated after the fault onset event.

Distributions can be continuous functions or discrete, and as usual, care must be afforded with numerical integration techniques in the case of Dirac and Kronecker deltas.

2.3. Part Parameters – Failure, Fault and Aging Effects

Parts exhibit a number of behaviors with respect to time including aging effects, faults, and failures. These effects are expressed in terms of the part parameters, which represent the role of the part within the system. For example, a capacitor can be modeled as having the parameters of capacitance, series resistance, and parallel conductance. Over the life of the capacitor, the capacitance can decrease dielectric degradation. These parameters affect the performance of a system with a capacitor. Failure effects, for example failing open or short, can be described in the part parameters or a new dynamic model without the capacitor can be used.

Part parameters vary in accordance with 4 effects:

1. Operating environment effects (temperature and pressure) – simulated with the system dynamics.
2. Aging effects – small effects as a result of the gradual accumulation of life.
3. Fault effects – accumulation of life becomes manifest in the part parameters in a more drastic manner.
4. Failure effects – catastrophic step change in part parameters.

Operating environment effects can be included in dynamic models based on deviations from a set of nominal parameters for states conditions.

Aging effects are usually the result of slow processes, long term usage and storage without incident. These can be

described as a function of a life state. Arrhenius approaches have been taken (Kuehl, 2010) in estimating variation of resistance and this is compatible with our approach. If the part parameter variations must be expressed purely as a function of time, then an element of the life state vector corresponding to a unity rate of life accumulation can be used; that is, the part representation has a built in clock. For example, for parts where there is no known correlation between failure and applied stress, but failures are distributed as a function of time, this behavior can be described with a stress factor equal to one.

The representation of faults is an extension of the method for representing the effects of aging. Faults are the manifestation of accumulated stress in the part parameters that occurs after a fault onset event. The occurrence of a fault onset event can be described using the same method as for describing failures through the use of distributions. For example, a part accumulates stress into a life state and demonstrates the effects of aging, after the occurrence of a fault onset event, the part parameters vary according to the fault behavior as a function of the still accumulating life state.

Failures are typically represented as the termination of aging and fault behaviors resulting in the part parameters taking a set of final values as determined by the failure mode. A part may have many failure modes, each corresponding to a particular set of part parameters.

2.4. Support for Physics of Failure Techniques

Modeling underlying parameters – the parameters used to represent the part in the dynamics are dependent on some underlying physical parameter. This is in keeping with the systems integration approach as it allows for definition of parts with internal behaviors – there is scope for self-similar, systems of systems model architectures. There is no fundamental limit to the level of detail that can be included in the mechanics of the through-life behaviors, although computational burden may establish practical bounds.

2.5. Stochastic Modeling with Random Walks

There is considerable literature content on the use of Markov chain and other random walk processes to model the progression of a part from full health through fault to failure. The accumulation of stress into a life state can be considered in similar terms. By use of stochastic integral techniques, random behavior can be modeled in continuous and discrete time and state.

The accumulation properties accumulation rate and accumulation severity have been defined. Accumulation rate is the probability that stress in any given time step will be accumulated into the life state. The accumulation severity is a gain factor that is applied to accumulated stress.

If the accumulation severity is the reciprocal of the accumulation rate, then in the limit as time tends to infinity, the average rate of stress accumulation is equal to the standards based definition. The accumulation of a life state is illustrated in Figure 1. It follows that the standard can still be applied, whilst permitting the expression of stochastic fault progressions. Taken in combination with the ability to describe physics of failure behaviors in the part parameters, the framework provides a strong basis for including models of different types in a single simulation environment.

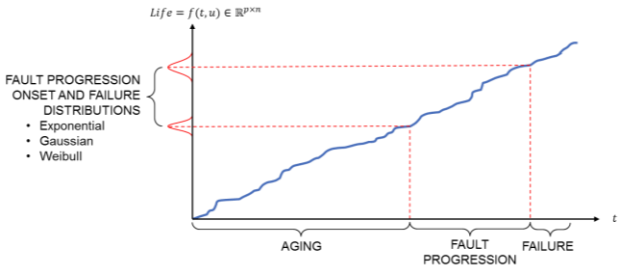


Figure 1. Accumulation of a "Life State"

2.6. System Representations

System representations must be extended to include the reliability and life data necessary to run simulations of models on product life timescales. In the framework, systems are described as a collection of parts. A system has the following attributes:

- A set of parts
- A dynamic model
- A set of specifications
- A set of use-cases

In typical time-domain simulations, a single part may only be represented by a single parameter (e.g. resistance). Part descriptions in this application are considerably more involved and should contain:

- A set of part parameters (observable and latent)
- A stress factor definition
- A set of life state accumulation properties
- Aging functions
- Fault onset distributions and fault effects
- Failure mode distributions and effects

2.7. Simulation Overview

The simulation uses parallel and sequential Monte Carlo approaches. The sequential part simulates a single instance of a system, and the variations that may occur over the life of that system. These variations can be internal variations in

part parameters or external factors like usage characteristics or operating environment. The parallel part of the Monte Carlo allows for variation in the initial conditions, which may be limited to the seed for random number generation or may include manufacturing tolerance or build configurations (which may include nominally identical systems that have differing part replacement histories). The steps in the simulation loop used in the sequential Monte Carlo are shown in Figure 2.

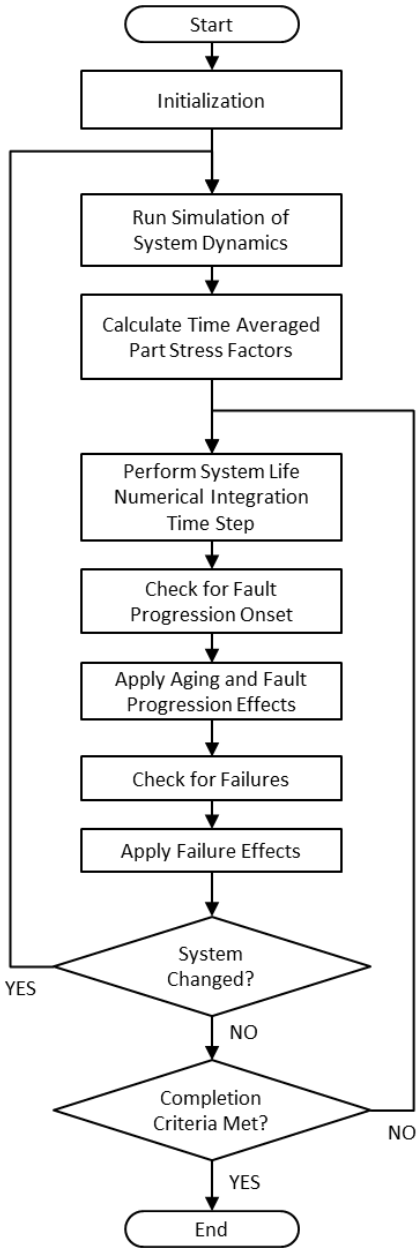


Figure 2. Simulation Steps

The numerical methods employed in running the simulation reuse and reinterpret the time series results from the simulation of the system dynamics in determining the amount of stress and life accumulated by the system. Only when the system is deemed to have changed sufficiently are the dynamic responses of the system re-simulated.

2.8. Specification Expression and Evaluation

In the assessment of robustness, aging and failures are simulated. The performance of the system is determined by measurement of some system properties against a set of rules. These properties can be time-domain simulation results, frequency domain transformations of simulation results or expressions formed from the set of available part parameters. A specification in the context of the framework is defined as:

<expression><operator><value><condition>

Where the expression contains the abovementioned system properties, the operator is a relational operator {=, ~=, >, >=, <, <=}, the value is a numeric or Boolean constant (but can also be another expression) and the condition is a constraint on the evaluation of the specification (evaluate subject to X being true, for example).

2.9. Use Cases

Use cases are the inputs to the dynamic model which indicate how the system is used. Each of these can be given a weighting, or in a more elaborate scheme, a usage sequence or schedule can be used over the life of the system. The set of use cases should describe in a complete sense the ways that the system will be used and the loads that the system will experience. Representations of the operating environment and ambient temperatures have been included.

2.10. Prognostics

By using techniques that take measurements of part parameters, either directly or by inference from system dynamic states or other parameters, prognostics aims to predict the time remaining before the system (or a part thereof) reaches the end of its life. Given the nature of the random behaviors incorporated into the simulation of system lives, and the nature of the inference algorithm, this prediction will have inaccuracies which can be classed as type I or type II errors:

- *Type I (False positive) error:* Prognostics falsely indicate imminent failure, system taken out of service to avoid failure effects resulting in a period when specifications are not met.
- *Type II (False negative) error:* Prognostics fail to indicate imminent failure, failure effects occur as they would have without prognostic.

2.11. Reliability, Robustness and Prognostics Assessment

A feature of the method is the comparative assessment of reliability, robustness and prognostic efficacy. Given the inclusion of fault and failure behavior, sets of system specifications and available prognostic techniques, the simulation results will indicate:

- the distribution of failures in time and their effects (a reliability analysis)
- the performance of the system with regard to the specifications over the range of part parameters (a robustness analysis)
- rates of false positive and false negative errors for the prognostic technique

3. AN RLC EXAMPLE

A resistor-inductor-capacitor (RLC) circuit serves to demonstrate clearly the essential features of the framework, without the distractions of a complex system. This example was chosen for its simplicity and for the fact that it calls out readily programmable sections of MIL-HDBK-217. The specifications and part parameters were selected arbitrarily, but such that simulation times were short. The inclusion of a thermal model was important for demonstrating coverage of a range of the stress factors. This example is not for the purpose of offering insight into the behavior of RLC circuits; the objective is to illustrate the incorporation of reliability behaviors in a time-domain robustness simulation. This example demonstrates the type of output data available and the reader is encouraged to envisage potential applications of the technique.

3.1. The System

The system was modeled using MATLAB/Simscape. Joule heating of each element was used in the thermal model. The thermal model was represented as a Cauer topology equivalent circuit. To enable calculation of the part stress factors, the model was required to output voltage, current and temperature time series. A schematic of the system is shown in Figure 3.

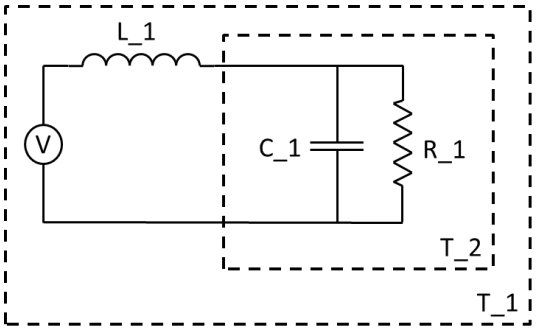


Figure 3. RLC circuit with thermal casings

3.2. Parts

Each part had a set of properties, parameters, aging functions and failure modes. Each part had exponentially distributed failure modes of *open* and *short*.

3.3. The Specifications

The specification applied to the circuit referred to the -3dB crossover frequency, which was calculated from the part parameters. The upper and lower limits for this frequency were defined as 2.340 and 2.436 radians per second, respectively.

3.4. System Usage

The use-cases for the model included sinusoidal and square wave input time series, and a range of ambient temperature and operating environment profiles.

3.5. Results

The results shown here are from a parallel Monte Carlo simulation where no variation was applied save for the random number generator seed. One hundred instances of the system were simulated with identical initial conditions and no through-life variation applied to the usage.

Figure 4 shows the variation in the characteristic frequency of the circuit as calculated from the part parameters. The vertical spikes are variations due to failure of a part - the distribution can be observed to be the result of constant hazard rate failures. The shaded regions correspond to the specification limits. There are breaches of the upper specification limit due to the aging of the parts.

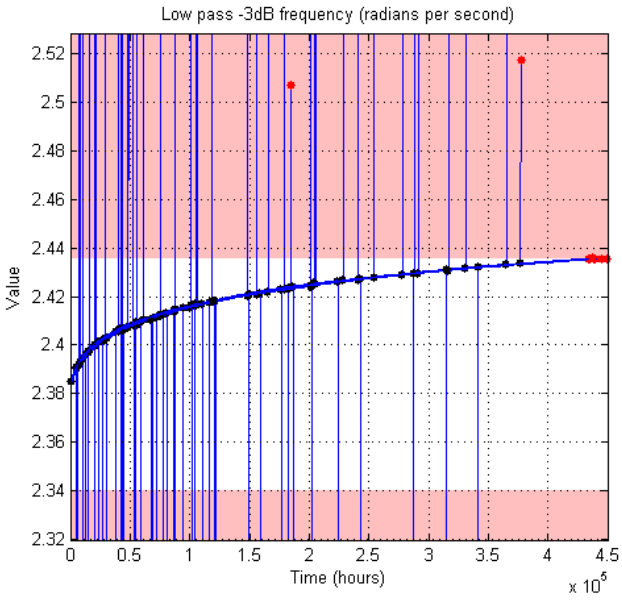


Figure 4. Through-life variation of frequency response characteristics

A selection of life states are shown in Figure 5. The randomized accumulation can be seen in the traces. The distributions of fault onset and failure are not representative as these life states were chosen for clarity of the graph.

The key aspect to these results is not the prediction regarding the reliability or robustness of the circuit, but that these data are the outputs of the same simulation.

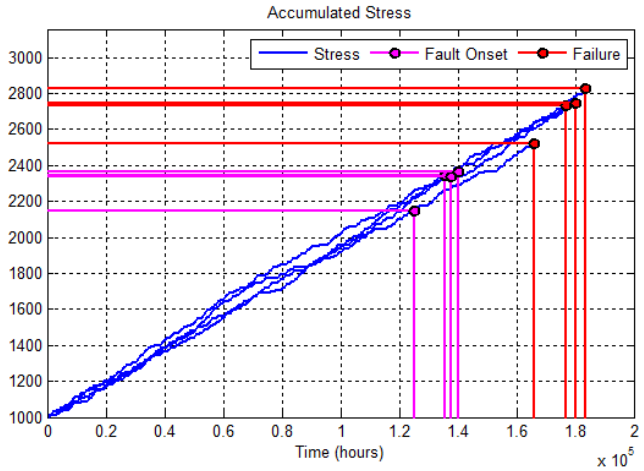


Figure 5. A subset of accumulated stress profiles

4. DISCUSSION

The results show the connection between simulation of the system dynamics, failures in the system and the adherence to the specifications for the system. The introductory example shows the type of outputs available using the framework; an enhanced demonstration would show the impact of variation of usage and operating environment on the reliability and robustness characteristics.

This following addresses advantages and disadvantages of the approach; it identifies key beneficial features and highlights areas which present new challenges in light of the novel techniques.

Advantages:

The incorporation of multiple types of part description into a model that captures causal relationships in a system yields an approach that can unify the analysis of a system design. This allows for trades between features of designs that were previously assessed by disparate means; reliability and robustness in particular. The unified analysis is well suited for complex systems. Application of variation in usage, operating environment and internal system states can yield variation in the reliability performance of the system and dominant system failure characteristics.

Models assessed against encoded specifications (and requirements) permits a closed loop design verification and validation methodology. Specification adherence in the face of applied variation forms the basis for an assessment of robustness. It can be argued that if the system design remains within the specification in the event of a failure, then the risk associated with the failure is mitigated by means of robustness. By the inclusion of the causal relationships of system parts, the impact of the long term presence of undetected degradations and failures to other elements of the system can be assessed. For example, if part A fails but the system remains in specification in the immediate term, is the long term performance of the system impacted due to increased stress on part B?

Further benefits are anticipated if this approach were coupled with executable specification modeling. This would permit early lifecycle design validation.

Other Considerations:

There is potentially a high computational expense of Monte Carlo simulations. Typical parallelization mitigations apply, but there are other mitigations that may yield substantially beneficial performance results:

- A database containing results for individual subsystems or units could allow for storage and reuse of costly simulation results.
- The consistent approach to modeling the many different types of behavior means that the execution of the simulation can be highly optimized.

It is recognized that the approach sets a high requirement for a large quantity of data about the parts of the candidate design. This may be offset with the development of libraries of standard parts, such that a design tool could make satisfaction of this requirement less challenging. Object oriented approach supports development of a library based design tool.

There is also a substantial outstanding burden to validate the approach against real world data, existing models and results from accelerated life testing. To that end, the use of the part stress approach is intended to be mathematically consistent with data in the standard, but the approach is not limited to standards based approaches. In the spirit of reliability standards, the methods demonstrated here are for the purpose of directing the attention of system designers at a stage where design decisions are critical.

Certain types of system may not be suitable for this approach and further work is required to determine the limits of applicability of the methods described here. Chaotic behavior, where the system state trajectories are highly sensitive to small deviations and variations from nominal can be simulated, however the computational burden may be well beyond reasonable limits.

5. CONCLUSION

An analytical framework to support systems-level decisions for robust performance has been presented. The "life state" method for time-domain simulation of unreliable systems has been explained. The methodology allows trade-space analysis on the appropriate use of prognostics to minimize the Size Weight and Power (SWaP) of redundant systems that otherwise would be needed. Significant potential benefits have been highlighted, yet further work is required to enhance demonstrations of the techniques described. It is anticipated that the development of these ideas will allow for better optimized designs, more unified analyses and a common approach to the design of reliable, robust and prognostic enabled systems.

ACKNOWLEDGEMENT

This work has been funded by the Air Force Research Laboratory (AFRL) through the Small Business Innovation Research (SBIR) program. Brett Jordan, AFRL technical point of contact, provided great support and encouragement during the course of this work.

NOMENCLATURE

CDF	cumulative distribution function
DAE	differential and algebraic equations
MTTF	mean time to failure
PDF	probability density function
RLC	resistor-inductor-capacitor

REFERENCES

Department of Defense (1995), Military Specification (MIL)-HDBK-217F NOTICE 2, *Reliability Prediction of Electronic Equipment*

Zio, E., (2009) Reliability Engineering: Old Problems and New Challenges, *Reliability Engineering and System Safety* **(94),** pp. 125-141.

Siu, N., (1994) Risk Assessment for Dynamic Systems: An Overview, *Reliability Engineering and System Safety* **(43),** pp. 43-73.

Kuehl, R. W., (2010) Using the Arrhenius Equation to Predict Drift in Thin Film Resistors, *CARTS Europe 2010 – 22nd Annual Passive Components Symposium*, (pp 121-133), November 10-11, International Congress Center, Munich, Germany.

BIOGRAPHIES

Nicholas A. Lambert, MEng, graduated from the University of Sheffield (UK) in 2006, with a first class honors degree in Mechanical Systems Engineering. He has keen interests in modeling, simulation & control as well as systems engineering. His experience includes modeling of physical systems, development for safety critical flight systems (BAE SYSTEMS – F-35 Fuel System IPT) and medical mechatronics design (with Imperial College London). Mr. Lambert's work has been centered on modeling fault progressions, failures, and prognostic algorithms, and has included development of design tools for assessment of reliability and robustness of dynamic systems with MATLAB/Simulink.

Kyle B. Ferrio earned the Ph.D. in Electrical Engineering and Computer Science from the University of Michigan at Ann Arbor. His research interests include topics in high-performance computing, algorithm development, and software modeling for complex and adaptive systems. His experience includes optimized design for optical fiber manufacturing, development of simulation codes for advanced optical systems and other computationally intense multi-physics modeling applications.

Douglas L. Goodman has an extensive background in electronic design, advanced diagnostics and prognostics, and test technologies. He has co-founded or managed other technology firms serving the industry, including Opmaxx (IC Test Software), Environmental Metrology Corporation (Process Metrology), and Analogy Inc. (Design Simulation). Doug received his BSEE from California Polytechnic State University, San Luis Obispo, and an MBA from the University of Portland and has held various managerial and engineering positions at firms including Tektronix and Honeywell.

Eddy Currents Signatures Classification by Using Time Series: a System Modeling Approach

Blaise Guépié[1], Mihaly Petreczky[2], and Stéphane Lecoeuche[3]

[1,2,3] *Mines Douai, IA, F-59508 Douai, France*
blaise.guepie@mines-douai.fr
mihaly.petreczky@mines-douai.fr
stephane.lecoeuche@mines-douai.fr

ABSTRACT

Non destructive testing methods are often used in order to detect and classify structural flaws. The detection of structural flaws is useful for maintenance. In this paper we propose to classify flaws in ferromagnetic materials by measuring Eddy currents. Our approach consists of two steps. First, we use a system identification algorithm to find a dynamical system which describes the data. Then, we use the parameters of this dynamical system as a feature vector and we use support vector machines in order to classify the various cracks. We test our method on a well-known benchmark.

1. INTRODUCTION AND MOTIVATION

Non destructive testing methods are used for checking the presence of structural flaws (cracks, deformations, etc.) of materials without causing damage. This is useful for predictive maintenance. The most important methods for non-destructive detection of structural flaws are the following: ultrasonic (Cantrell & Yost, 2001), acoustic emission (Madaras, Prosser, & Gorman, 2005), terahertz ray (Němec, Kužel, Garet, & Duvillaret, 2004), X-ray (Elaqra, Godin, Peix, R'Mili, & Fantozzi, 2007), thermal (Clark, McCann, & Forde, 2003), optical method (Jie, Siwei, Qingyong, Hanqing, & Shengwei, 2009) and eddy currents (EC) (Smid, Docekal, & Kreidl, 2005).

In this paper, we propose using Eddy currents for detecting flaws. Methods based on measuring Eddy currents are popular, because measuring Eddy currents is cheap and it allows detecting clogged defects and to classify cracks. For any classification method, feature extraction is one of the critical steps. For Eddy currents, several feature extraction methods exist in the literature. (Jo & Lee, 2009; Song & Shin, 2000; Liu et al., 2013) use maximum amplitude and phase angle or width of defect signal; (Oukhellou, Aknin, & Perrin, 1999; Smid et al., 2005) focus on the Fourier or wavelets transform parameters; (Lingvall & Stepinski, 2000; Ye et al., 2009) use principal component analysis or its kernel version.

In comparison to the existing methods for feature extraction, the main novelty of the proposed method lies in using parameters of dynamical systems as feature. This represents a novel application of system identification techniques to fault detection and health monitoring of ferromagnetic materials based on Eddy currents.

Our approach is based on two steps. First, using the measured data, we find a parametric dynamical model. This model represents current impedance values of eddy currents as function of past impedance values. We use a system identification algorithm for identifying the model parameters based on measured data. Thus, the obtained parameters serve as feature. We assume that each flaw corresponds to an unique parameter vector. We then use support vector machines to compute a classifier on the extracted parameter space.

The experimental evaluation shows that our approach gives good results.

The paper is organized as follows. Section 2 is devoted to the problem statement. Section 3 presents the data pre-processing step including denoising and re-sampling. A new method of feature extraction based on dynamical systems identification is explained in Section 4. The Support Vector Machines operating is briefly described in Section 5. Section 6 shows an example of classification of flaws using Eddy currents . Some conclusions are drawn in Section 7.

2. PROBLEM DEFINITION

Eddy currents are used in many applications of non destructive testing. When a conductive material is within a time-variable magnetic field created by a coil subjected to an alternative current, induced Eddy currents are developed inside the material without altering its characteristics. When an in-

Blaise Guépie et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

homogeneity, a change in geometry or a flaw is present in the material, variations in the phase and magnitude of these eddy currents can be monitored, as they lead to a change of the coil impedance. This is the principle of material inspection by Eddy currents. Sensors travel across the surface of the material and the variations of the coil impedance are acquired and compared with an impedance reference. Then, in presence of a crack, the impedance data varies as function of sensor or material displacement, following a trajectory into a complex plane where abscissa is the resistance and the reactance is the ordinate.

By firstly considering aluminum structures, the goal of the method presented in this paper is to propose to automatically classify the type of defaults or cracks using as input the impedance data trajectory. To present the method, an existing database [1] composed of Eddy currents signatures from aluminum aircraft structures has been used. The database is composed of twelve types of crack, characterized by both penetration angle into the material and depth of penetration. Figure 1 shows characteristics of all the twelve cracks. For example, the first crack type is defined by $1.5mm$ of depth and $90°$ of penetration angle.

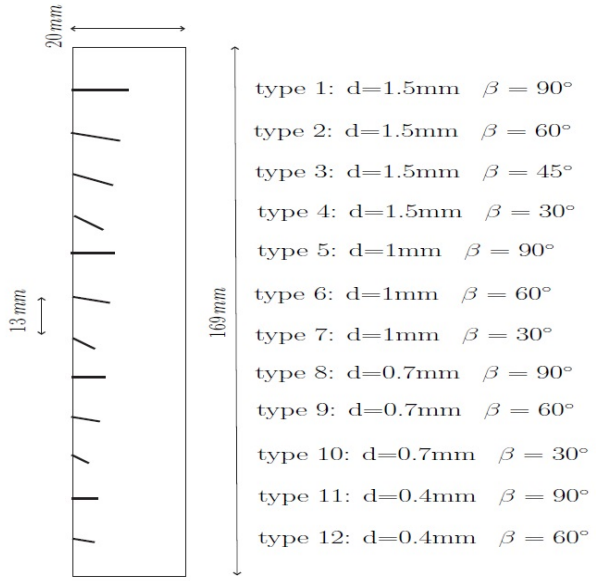


Figure 1. Aluminum sample with machined notches of different penetration angles and depths.

For each crack, acquired impedance data are complex discrete time series

$$\tilde{z}(k) = \tilde{x}(k) + j\tilde{y}(k),$$

where the resistance curve $\tilde{x}(k)$ and the reactance curve $\tilde{y}(k)$ are known. Each type of crack is scanned 20 times by a coil. This leads to 20 impedance trajectories. Figure 2 illustrates different impedance trajectories for several types of crack.

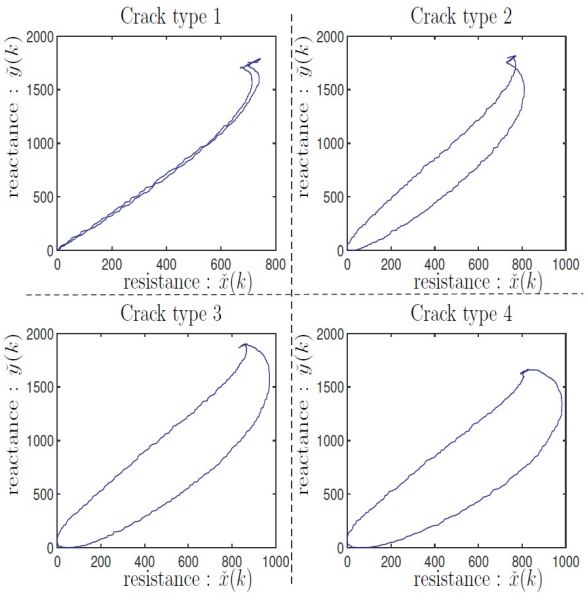


Figure 2. Impedance trajectories of four crack types.

The originality of this paper lies in classifying crack types using the parameters of temporal models that fit the impedance trajectories. Each component of the trajectory (resistance and reactance) will be considered as time series ARX model where its parameters will be used to classify the crack type. So, the main steps of the proposed method are :

- from each inspection, extract sequences $\tilde{x}(k)$ and $\tilde{y}(k)$,
- estimate θ_x and θ_y the parameters associated to the time series models,
- knowing the set of θ_x and θ_y, corresponding to the whole inspection database, build the classifiers.

Before explaining in detail the estimation and the classification steps, it is worth introducing remarks on the data preprocessing step.

3. DATA PREPROCESSING

First, in order to reduce noise impact, resistance $\tilde{x}$ and reactance $\tilde{y}$ are filtered. A standard median filter is used. For $k \geq 1$, the median values
$\check{x}(k)$ of $\{\tilde{x}(k - L_x + 1), \cdots, \tilde{x}(k + L_x - 1))\}$ and
$\check{y}(k)$ of $\{\tilde{y}(k - L_x + 1), \cdots, \tilde{y}(k + L_x - 1))\}$ respectively replace $\tilde{x}(k)$ and $\tilde{y}(k)$ where $2L_x - 1$ end $2L_y - 1$ are the prescribed length of the median filter window.

Moreover, collected data come from a manual scanning. This implies that the scan speed varies over time and variations impact the impedance curve shape for the same crack. To reduce the effect of variate scanning speed, data $\{\check{x}(k)\}_{k=1}^{M}$ and $\{\check{y}(k)\}_{k=1}^{M}$ are re-sampled in order to get the same number

[1] freely available on the website : http://wireless.feld.cvut.cz/diagnolab/node/16

of points for each crack.These filtered and re-sampled data $\{\hat{x}(k)\}_{k=1}^{N}$ and $\{\hat{y}(k)\}_{k=1}^{N}$ will serve as the second learning dataset.

Each type of cracks is characterized by the following two properties: "depth" and "angle". Depth refers to the depth of the crack, and angle refers to the angle formed by the crack and the horizontal axis. That is, each crack is identified with a pair of numbers $(depth, angle)$, where $depth$ denotes the depth of the crack and $angle$ denotes the angle of the crack. Hence, we can classify cracks as follows. First, we construct a classifier which determines the angle associated with each crack based on measurement data. In this way, we obtain several groups of cracks, each group representing cracks with the same angle. Second, for each angle α, we construct a classifier which determines the depth of a crack whose angle equals α. This classifier will use measurements to determine the depth. Note that the second classifier is supposed to distinguish only cracks with the same angle. Figure 3 shows curves of impedance values for cracks with $60°$ af angle and $1.5mm$, $1mm$, $0.7mm$, $0.4mm$ of penetration depths. Each curve is a finite collection $\{(\hat{x}(k), \hat{y}(k))\}_{k=1}^{N}$ of data points, where $\hat{x}(k)$ is the real part and $\hat{y}(k)$ is the complex part of $\hat{z}(k)$, the filtered and re-sampled impedance value measured at time step k. It can be seen from the data that the shapes of the four cracks are similar but some are larger than other. Therefore, we can assume that the depth does not have impact on the shape of the curve except its magnitude. For this reason, we will apply a normalization step in order to find crack angles. That is to say, before computing the first classifier, we will divide the data points by a constant. Thus, the first dataset is composed of

$$\{(x(k), y(k))\}_{k=1}^{N} = \left\{ \frac{(\hat{x}(k), \hat{y}(k))}{\max\limits_{k\in\{1,\cdots,N\}} \sqrt{\hat{x}^2(k) + \hat{y}^2(k)}} \right\}_{k=1}^{N}$$

. Note that for the computation of the second classifier, we will use the original data points $\{(\check{x}(k), \check{y}(k))\}_{k=1}^{N}$, since the crack's depth influences the magnitude of the data points and thus normalization could lead to loss of information.

4. FEATURE EXTRACTION BASED ON DYNAMICAL SYSTEMS

As we have seen before, each crack observation is composed of time series data points arising from Eddy currents measurements. It can be represented as curve in the complex plane, since each impedance value is a complex number. However, such a representation discards the temporal dependence between various data points. For this reason, in order to compute classifiers, we will use the time series $\{x(k)\}_{k=1}^{N}$ and $\{y(k)\}_{k=1}^{N}$. We will use these time series to compute a dynamical system whose input-output behavior is consistent with them. We assume that each group of flaws (i.e, each angle

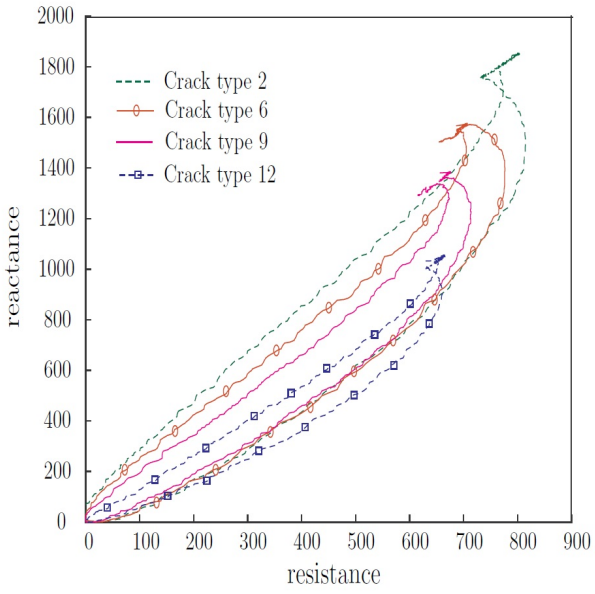


Figure 3. Cracks with $60°$ of angle and four penetration depths.

of penetration) determines an unique pair of parameters vectors, where each parameters vector corresponds to a dynamical system. It is worth noting that each pair of parameters vectors is extracted from one crack observation. Thus, parameters vectors are independent from each over. The first dynamical system models the resistance:

$$x(k) = \theta_x^T \phi_x(k) + \xi_x(k), \tag{1}$$

and the second one models the reactance:

$$y(k) = \theta_y^T \phi_y(k) + \xi_y(k), \tag{2}$$

where
$\phi_x(k) = [x(k-1), \cdots, x(k-n_{a_x}), y(k-1), \cdots, y(k-n_{b_x}), 1]^T$,
$\phi_y(k) = \left[y(k-1), \cdots, y(k-n_{a_y}), x(k-1), \cdots, x(k-n_{b_y}), 1\right]^T$,
(n_{a_x}, n_{b_x}) and (n_{a_y}, n_{b_y}) are models orders,$\{\xi_x(k)\}_{k\geq 1}$, $\{\xi_y(k)\}_{k\geq 1}$ are independent identically distributed random sequences and θ_x, θ_y are the associated parameters vectors.

We assume that for each crack, the pair of parameters vectors (θ_x, θ_y) determine the angle of the crack. More precisely, we assume that these pairs are close when cracks belong to the same group, i.e, the penetration angle is the same.

In this paper, we will use the *Recursive Least-Squares method (abbreviated by RLS)* as linear system identification algorithm because its recursive form is exact comparatively to the Least Mean Squares (LMS) and then, it converges more quickly to the solution. The RLS algorithm was proposed to determine the parameter θ of the equation

$$w(k) = \theta^T \psi(k) + \xi(k) \tag{3}$$

from finitely many measurements $\{w(k), \psi(k)\}_{k=1}^{N}$. Note that both (1) and (2) are of the form (3) with a suitable choice of $\theta(k)$, $\psi(k)$, $\xi(k)$ and $w(k)$. Below we describe the RLS algorithm. We will follow the presentation of (Ljung & Söderström, 1983). Let I be the identity matrix and let $P(0) = \sigma I$ be the initial auto-correlation matrix of data. For each new observation $(w(k), \psi(k))$, the update of $\left(\hat{\theta}^T(k-1), P(k-1)\right)$ is given by:

$$\begin{aligned} \xi(k) &= w(k) - \hat{\theta}^T(k-1)\psi(k), \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + \frac{\xi(k)P(k-1)\psi(k)}{\alpha + \psi(k)^T P(k-1)\psi(k)}, \\ P(k) &= \alpha^{-1}P(k-1)\left[I - \frac{\psi(k)\psi(k)^T P(k-1)}{\alpha + \psi(k)P(k-1)^T\psi(k)}\right] \end{aligned} \tag{4}$$

where $\alpha \in [0, 1]$ is the forgetting factor.

5. CLASSIFICATION OF CRACKS

The goal of this part is to identify the class membership of each crack. The classification is in two step. The first one is for discriminating the penetration angle into the material and the second one is for selecting the penetration depth. Several methods exist in classification theory. For labeled data, the Support Vector Machines (SVM) developed in (Vapnik, 2000) achieve excellent performance according to (Caruana & Niculescu-Mizil, 2006; Khelil, Boudraa, Kechida, & Drai, 2005). The following lines briefly explain SVM principle.

5.1. Two classes SVM

The SVM has been firstly used to separate two classes. Its principle is to maximize the separation margin ; the margin being the distance between the closest observations and the separator.

Consider a given training set $\{x_k, y_k\}_{1 \leq k \leq N}$ where the observation $x_k \in \mathbb{R}$ and the class variable $y_k \in \{-1, 1\}$. Suppose that data are linearly separable, i.e, there exists a linear classifier (w, b) such as

$$\begin{cases} w^T x_k + b \geq +1 & \text{if} \quad y_k = +1 \\ w^T x_k + b \leq -1 & \text{if} \quad y_k = -1 \end{cases}. \tag{5}$$

The problem of finding the separator which maximizes the margin is equivalent to :

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\langle w, w\rangle \\ \text{constraint to} \quad & y_k(w^T x_k + b) > 1,\ 1 \leq k \leq N, \end{aligned} \tag{6}$$

where $\langle , \rangle$ is the dot product.

Generally, data are not separable. In this case, the margin of some observations are allowed to be less than one. Slacks variables $\epsilon_k \geq 0$, $1 \leq k \leq N$ are introduced in order to solve the problem. The optimization problem becomes

$$\begin{aligned} \min_{w,b,\epsilon_k} \quad & \frac{1}{2}\langle w, w\rangle + C\sum_{k=1}^{N}\epsilon_k \\ \text{such that} \quad & \begin{cases} y_k(w^T x_k + b) > 1 - \epsilon_k \\ \epsilon_k \geq 0 \end{cases} \quad \text{for } 1 \leq k \leq N, \end{aligned} \tag{7}$$

where C is a positive real constant for determining the tolerance of the SVM to the poorly separated observations. The solution are $w = \sum_{k=1}^{N}\gamma_k y_k x_k$ and $b = b_0$ where $\gamma_k \geq 0$ for $1 \leq k \leq N$ and b_0 are obtained by solving the dual formulation of (7). Then, the decision function is

$$f(x) = sign\left(\sum_{k=1}^{N}\gamma_k y_k \langle x_k, x\rangle + b_0\right). \tag{8}$$

When datasets are linearly non-separable, the trick is to project them on a high-dimensional feature space by using a nonlinear map $\psi(\cdot)$ such as the projections are linearly separable. Thanks to the Mercer's condition (Mercer, 1909), the calculation of the dot product $\langle \psi(x_{k'}), \psi(x_k)\rangle$ which often requires a lot of computational resources is replaced by the calculation of the kernel $K(x_{k'}, x_k)$. Two kernels are widely used: the Gaussian kernel $K(x_{k'}, x_k) = \exp\left(-\frac{\|x_{k'} - x_k\|^2}{2\sigma^2}\right)$ and the homogenous polynomial kernel $K(x_{k'}, x_k) = \langle x_{k'}, x_k\rangle^d$, where σ an d are tuning parameters.

In this paper, these parameters are selected as those minimize the leave-one-out cross validation error whose the procedure consists of:

- splitting the data set of size k into k smaller subsets
- a model is trained using k-1 subsets as training data
- the resulting model is validated on the remaining subset
- the previous both lines are repeated k times.

After using a kernel, the decision function (8) becomes

$$f(x) = sign\left(\sum_{k=1}^{N}\gamma_k y_k K(x_k, x) + b_0\right). \tag{9}$$

5.2. Multi-class SVM

The mutli-class SVM is an extended version of two classes SVM. Here, it is supposed that the number of classes is greater than two.

Here, the One Against One SVM is used for classifying more than two classes. This approach is very intuitive. It consists of making several classifiers in order to compare pairs classes.

Thus, for classifying a dataset between M classes, we need to make $\frac{M(M-1)}{2}$ separators (Moreira & Mayoraz, 1998). A majority vote across the classifiers is applied to classify a new observation.

In brief, the global crack classification procedure is given by algorithm 1.

6. EXPERIMENTAL RESULTS

In order to test the reliability of our method, a database [2] composed of Eddy currents signatures from aluminum aircraft structures is used. In this database, there are twelve types of crack. Each type of crack is characterized by the angle and the penetration depth, is recorded 20 times. Figure 1 shows characteristics of all the twelve cracks.

The first classification task is devoted to the penetration angle. Four groups are created from the twelve types of crack. The first group contains 80 cracks with 90° of angle and 1.5, 1, 0.7, 0.4 mm of penetration depth. The second group contains 80 cracks with 60° of angle and 1.5, 1, 0.7, 0.4 mm of penetration depth. In the third group, there are 20 cracks with 45° of angle and 1.5 mm of penetration depth. The last group contains 60 cracks with 30° of angle and 1.5, 1, 0.7 mm of penetration depth.

The parameters used in the first part of classification algorithm (see algorithm 1) are the following. The median filters widows size are $L_x = 30$ observations and $L_y = 30$ observations. The fixed number of observations is $N = 300$. The re-sampling factor is $factor = N/M$ where M is the curve number of observations. The Recursive Least Squares forgetting factor is $\alpha = 9.99 \times 10^{-1}$, its value for the initial auto-correlation matrix is $\sigma = 10$ and its initial parameters $\theta_x(0)$, $\theta_y(0)$ are randomly selected.

Each crack observation is composed of time series data points arising from Eddy currents measurements. For discriminating different cracks, our procedure consists to extract representative parameters vector from each crack observations. The extracted parameters vectors are independent from each over. Several values of order are tested for both dynamical systems identification (1) and (2) in order to select the model parameters. The values which minimize the least squares errors are $(n_{a_x}, n_{b_x}) = (0, 2)$ and $(n_{a_y}, n_{b_y}) = (0, 2)$. Figure 4 shows estimations of measured resistance and reactance. Both estimated curves obtained from two order dynamical systems are close to real curves.

It is worth nothing that the goal of our extracted parameters is not exactly to predict the real curves of resistance and reactance. The goal of prediction is to evaluate whether the extracted parameters vector explains the dynamic of the considered time series. The evaluation step allows defining the

[2] Free available on the website : http://wireless.feld.cvut.cz/diagnolab/node/16

Algorithm 1 Procedure of cracks classification

1: Have Eddy currents measuring data
2: Removal of edge data: a set of uninformative data (null data) to the right of each curve is deleted
3: Data filtering: set the windows sizes L_x and L_y of median filters
4: Duplication of the dataset $\{(\check{x}(k), \check{y}(k))\}_{k=1}^{M}$: the 1st dataset will be processed for discriminating angle and the 2nd dataset will be used for discriminating depth without further processing
5: Re-sampling of resistance and reactance curves extracted from the 1st dataset : set the re-sampling factor $factor$ in order to obtain the same number of observations N in each curve; $\{(\hat{x}(k), \hat{y}(k))\}_{k=1}^{N}$ is the obtained dataset.
6: Normalization: resistance and reactance curves extracted from the 1st dataset are divided by the magnitude:

$$\{(x(k), y(k))\}_{k=1}^{N} = \left\{ \frac{(\hat{x}(k), \hat{y}(k))}{\max\limits_{k\in\{1,\cdots,N\}} \sqrt{\hat{x}^2(k) + \hat{y}^2(k)}} \right\}_{k=1}^{N}$$

7: Extraction of dynamical system parameters vectors (θ_x, θ_y) from $\{(x(k), y(k))\}_{k=1}^{N}$ and the Recursive Least Squares algorithm:
- set the forgetting factor $\alpha \in [0, 1]$, the value σ for the initial auto-correlation matrix of data $P(0) = \sigma I$, and $(\theta_x(0), \theta_y(0))$ the initial parameters vectors
- **for** $i \in \{1, \cdots, N\}$ **do**

$$\xi_x(k) \leftarrow x(k) - \theta_x^T(k-1)\phi_x(k),$$
$$\theta_x(k) \leftarrow \theta_x(k-1) + \frac{\xi_x(k)P(k-1)\phi_x(k)}{\alpha + \phi_x(k)^T P(k-1)\phi_x(k)},$$
$$P(k) \leftarrow \frac{P(k-1)}{\alpha}\left[I - \frac{\phi_x(k)\phi_x(k)^T P(k-1)}{\alpha + \phi_x(k)P(k-1)^T\phi_x(k)}\right]$$

- **end for**
- find θ_y as in the above lines

8: Using a multi-class Support Vector Machines for determining angle:
- set θ the concatenation of θ_x and θ_y
- the dataset of θ is separated in two parts. the first part will be used for training and the second part for evaluation algorithm performance
- set the kernel, the kernel parameter and the positive constant of tolerance C

9: Using a multi-class Support Vector Machines and the 2nd dataset $\{(\check{x}(k), \check{y}(k))\}_{k=1}^{M}$ for determining the depth among depths belonging to the previously found angle group:
- calculation of the magnitude mg :

$$mg = \max_{k\in\{1,\cdots,M\}} \sqrt{\check{x}^2(k) + \check{y}^2(k)}$$

- the subset of the 2nd dataset containing magnitudes of the previously found angle is used for training and the second part of data used in step 8 is used for the evaluation
- set the kernel, the kernel parameter and the positive constant of tolerance C

quality of extracted feature so that this one could be used to the second step. The assumption is that whatever the type of crack, the used model structure is fixed. Then, only the parameter of the identified model will be used to characterize the type of crack.

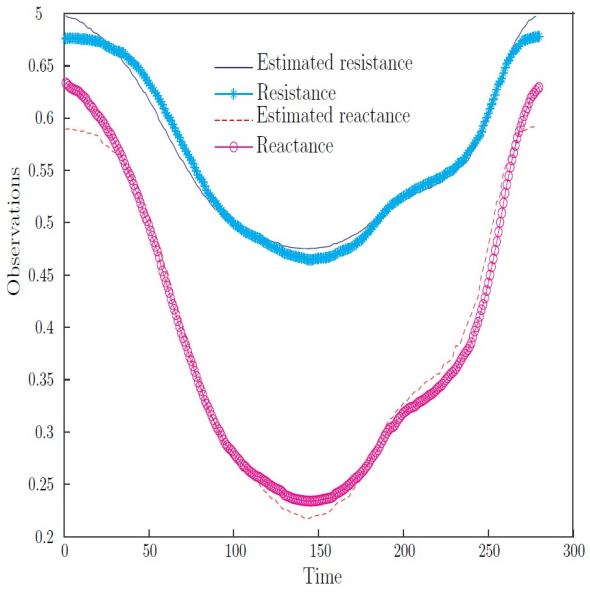


Figure 4. Estimation of resistance and reactance from two order dynamical systems model.

For each crack data recorded, the pair of extracted parameters vectors (θ_x, θ_y) belongs to $\mathbb{R}^3 \times \mathbb{R}^3$. Figures 5 and 6 show these parameters in two three-dimensional spaces. It can be seen that the four groups are well separated. Before classifying, parameters θ_x and θ_y are concatenated. The new vector from this concatenation θ belongs to $\mathbb{R}^6$.

The One Against One SVM with a Gaussian kernel is used for the classification. The pair (σ, C) of the kernel parameter and the constant of tolerance are searched into a grid $\left[10^{-5}, 10^{4}\right] \times \left[10^{-5}, 10^{4}\right]$ containing 100 equidistant pairs. The selected pair $(\sigma = 10, C = 10)$ is that minimizes the miss-classification error rate after using leave-one-out cross validation. The minimum of the miss-classification error rate is equal to 1.25%. Tables 1 shows the confusion matrix of the One Against One SVM. Thus, we can say that our approach has a good classification performance with respect to the penetration angle into material.

In order to evaluate the efficiency of our classification procedure, we are going to realize the classification with extracted parameters obtained by a Principal Component Analysis applied on Fourier descriptors (FD-PCA). The FD-PCA procedure consists firstly to extracted Fourier descriptors from our crack observations. Discrete Fourier descriptors are defined as

$$df(p) = \frac{1}{N} \sum_{k=1}^{N} z(k) \exp\left(-j2\pi p(k-1)/N\right),\ p = 1, \cdots, N$$

After the calculation of the discrete Fourier descriptors for each crack observation, the most representative descriptors are selected by using Principal Component Analysis. Three complex descriptors are chosen and these ones represent 93% of the total variance. By using One Against One SVM and the leave-one-out cross validation, the miss-classification error rate is equal to 7.5%. Tables 2 shows the confusion matrix of the classification based on the FD-PCA procedure.

It can be seen that our extraction procedure based on dynamical systems combined to SVM gives better results than the FD-PCA procedure combined to SVM for the same number of extracted parameters (6 real parameters for the first one and 3 complex parameters for the second one). Thus, our classification algorithm is efficiency and promising.

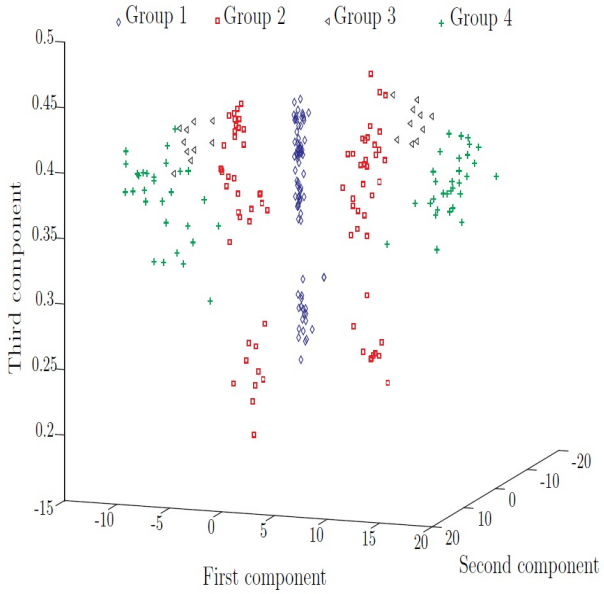


Figure 5. Illustration of the parameter θ_x for the four crack groups.

Table 1. Confusion matrix of the One Against One SVM.

	Predicted Group			
Actual Group	Group 1	Group 2	Group 3	Group 4
Group 1	80	0	0	0
Group 2	0	80	0	0
Group 3	0	0	19	1
Group 4	0	1	1	58

The second step of classification consists of discriminating the penetration depth for cracks with the same penetration

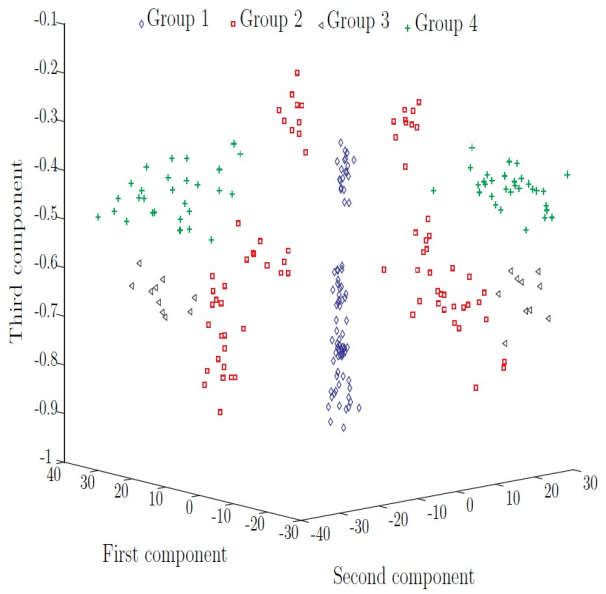


Figure 6. Illustration of the parameters θ_y for the four crack groups.

Table 2. Confusion matrix of the One Against One SVM applied on the FD-PCA feature extraction.

	Predicted Group			
Actual Group	Group 1	Group 2	Group 3	Group 4
Group 1	77	3	0	0
Group 2	8	71	1	0
Group 3	0	2	17	1
Group 4	0	1	2	57

angle. To do this, the second dataset, i.e, without normalization is used. Figures 7 shows the second classification step in group 1. Similar figures are obtained for groups 2 and 4. For cracks classified in group 3, there is not a second step because the penetration depth is unique ($1.5mm$). According to this figure, recorded data can be well separated by hyperplanes. The One Against One SVM with a Gaussian kernel is used for the three classifications in the second step. As we previously mentioned, the pair (σ, C) of the kernel parameter and the constant of tolerance are searched into a grid $\left[10^{-5}, 10^{4}\right] \times \left[10^{-5}, 10^{4}\right]$ containing 100 equidistant pairs. The selected pairs of kernel parameter and constant of tolerance of group 1, 2 and 4 are receptively $(\sigma = 1, C = 1)$, $(\sigma = 0.1, C = 0.1)$ and $(\sigma = 0.1, C = 0.1)$. The miss-classification error in step two, i.e for the three classifications is null. In other words, according to available data, if a crack is classified in the right group (ie, if the right angle is selected), the right depth is automatically selected. Hence, the global miss-classification error is equal to the first step miss-classification (approximately equal to 1%).

Remark.

The previous classification uses Gaussian kernel. However, when Gaussian kernel is replaced by homogenous polynomial kernel respectively with parameters $(d = 3, C = 10^{-3})$, the same miss-classification error is obtained. Our classification seems robust with respect to the selected SVM kernel.

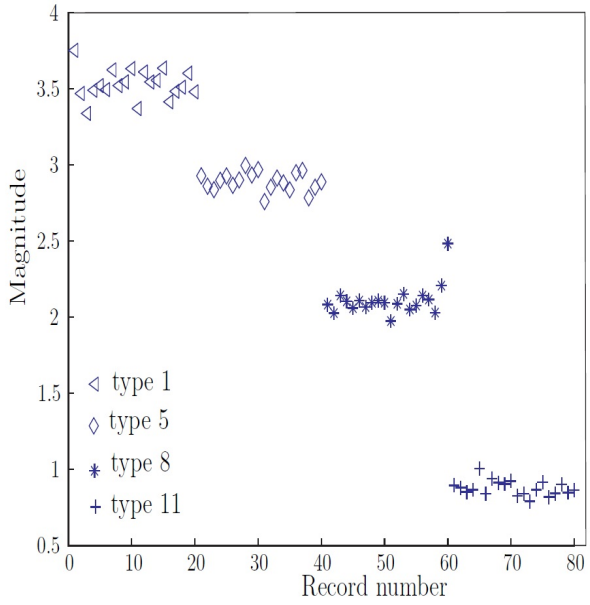


Figure 7. Impedance magnitude of the first group records.

7. CONCLUSION

This paper addresses the problem of classifying cracks by using measurements of Eddy currents. The paper proposes a new approach for cracks classification which uses dynamical systems. The parameters of these dynamical systems form the feature space. The parameters vectors are found from the measured data by using an algorithm from systems identification. The classification is done in two steps. The first one is to group cracks according to theirs penetration angles into the material. The second one is to group them according to penetration depths. Our method is evaluated on a particularly challenging benchmark, for which the cracks are more difficult to classify due to the variation of the scanning speed. After using multi-class SVM for both steps classification, the miss-classification error is approximately equal to 1%. This means that our approach is efficiency and promising.

REFERENCES

Cantrell, J. H., & Yost, W. T. (2001). Nonlinear ultrasonic characterization of fatigue microstructures. *Interna-*

tional Journal of Fatigue, *23, Supplement 1*(0), 487 - 490.

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on machine learning* (pp. 161–168). New York, NY, USA: ACM. doi: 10.1145/1143844.1143865

Clark, M., McCann, D., & Forde, M. (2003). Application of infrared thermography to the non-destructive testing of concrete and masonry bridges. *Ndt & E International*, *36*(4), 265–275.

Elaqra, H., Godin, N., Peix, G., R'Mili, M., & Fantozzi, G. (2007). Damage evolution analysis in mortar, during compressive loading using acoustic emission and x-ray tomography: Effects of the sand/cement ratio. *Cement and Concrete Research*, *37*(5), 703 - 713.

Jie, L., Siwei, L., Qingyong, L., Hanqing, Z., & Shengwei, R. (2009, July). Real-time rail head surface defect detection: A geometrical approach. In *Industrial electronics, 2009. isie 2009. ieee international symposium on* (p. 769-774).

Jo, N. H., & Lee, H.-B. (2009). A novel feature extraction for eddy current testing of steam generator tubes. {*NDT*} *& E International*, *42*(7), 658 - 663.

Khelil, M., Boudraa, M., Kechida, A., & Drai, R. (2005). Classification of defects by the svm method and the principal component analysis (pca). *World Acad. Sci. Eng. Technol*, *9*, 226–231.

Lingvall, F., & Stepinski, T. (2000). Automatic detecting and classifying defects during eddy current inspection of riveted lap-joints. {*NDT*} *& E International*, *33*(1), 47 - 55.

Liu, B., Hou, D., Huang, P., Liu, B., Tang, H., Zhang, W., ... Zhang, G. (2013). An improved pso-svm model for online recognition defects in eddy current testing. *Nondestructive Testing and Evaluation*, *28*(4), 367-385.

Ljung, L., & Söderström, T. (1983). *Theory and practice of recursive identification*. MIT press Cambridge, MA.

Madaras, E. I., Prosser, W. H., & Gorman, M. R. (2005). Detection of impact damage on space shuttle structures using acoustic emission. In *Review of progress in quantitative nondestructive evaluation: Volume 24* (Vol. 760, pp. 1113–1120).

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, *209*, 415–446.

Moreira, M., & Mayoraz, E. (1998). Improved pairwise coupling classification with correcting classifiers. In *Machine learning: Ecml-98* (pp. 160–171). Springer.

Němec, H., Kužel, P., Garet, F., & Duvillaret, L. (2004). Time-domain terahertz study of defect formation in one-dimensional photonic crystals. *Applied optics*, *43*(9), 1965–1970.

Oukhellou, L., Aknin, P., & Perrin, J.-P. (1999). Dedicated sensor and classifier of rail head defects. *Control Engineering Practice*, *7*(1), 57 - 61.

Smid, R., Docekal, A., & Kreidl, M. (2005). Automated classification of eddy current signatures during manual inspection. *NDT & E International*, *38*(6), 462–470.

Song, S.-J., & Shin, Y.-K. (2000). Eddy current flaw characterization in tubes by neural networks and finite element modeling. {*NDT*} *& E International*, *33*(4), 233 - 243.

Vapnik, V. (2000). *The nature of statistical learning theory*. Springer.

Ye, B., Huang, P., Fan, M., Gong, X., Hou, D., Zhang, G., & Zhou, Z. (2009). Automatic classification of eddy current signals based on kernel methods. *Nondestructive Testing and Evaluation*, *24*(1-2), 19–37.

Operation Condition Monitoring using Temporal Weighted Dempster-Shafer Theory

Xiaoyun Wang[1], Tingdi Zhao[2]

[1,2] *School of System Engineering and Reliability, BeiHang University, Bejiing,100191, China*

wangxys@gmail.com
ztd@buaa.edu.cn

ABSTRACT

System operation is a real time, dynamic decision process, a continuous observation should be implemented to support timely decision. Real time condition monitoring and diagnosis is featured with ongoing event sequence. The more recent observation, the much detailed, accurate information, and the more obsolete observations with much weak correlation to current faults and errors vise versa.

Dempster-Shafer evidence theory is best suitable for the problem of redundant sensors, insufficient data reasoning. However, D-S base applications largely focused on causational relationship between symptoms and effects, and the fusion process of evidences was performed regardless whatever order observed. As an improvement to the frame of discernment of the D-S theory, we purposed a time weighted evidence combination method. Observed events were extracted from multiple time points to form a temporal evidence sequence. Basic probability assignment was altered by temporal weights in accordance with the time proximity between the observed events and current time. The temporal weights value set was in accordance with its occurring time point. Evidences with same timestamps should be allocated with the same temporal weights. An example was discussed to illustrate the temporal weight, D-S rule based assessment framework. In the framework, latest observed evidences stream were combined into the framework to improving fault recognition.

1. INTRODUCTION

Condition assessment for system operation is a real time, dynamic decision process, during that course, a continuous observation should be implemented to support timely condition assessment. Currently, as a method widely in the area of fault diagnosis applications, Dempster-Shafer evidence theory is best suitable for the problem of redundant sensors, reasoning of insufficient data which might be imprecise and incomplete(Yang, 2006)(Parikh,2001)

As an extension of traditional probabilistic theory, the Dempster-Shafer Theory (DST) of evidence provides beneficial approaches to uncertain reasoning. In the network security area, DST was used as a method for incursion detection (Lan, 2010), intrusion prioritizing (Zomlot, 2011). In ubiquitous network and pervasive computing, DST was applied to recognize situation and activities in smart environment (McKeever, 2009). It also play an important role in bank fraud detection applications (Beranek, 2013). Some of these researches concerned the temporal property of evidence to improve performance of detection, for examples, McKeever tried to use a duration measure to generate the belief of event and evidence.

Our research focused on the problem of temporal aspect of DST evidence. During the online condition monitoring, some observed information might not up to date sufficiently while others may appears better timeliness. Outdated information as one of three kind of major information problems (Garvin, 1988), is not sufficiently for the task of fault detection. The more recent observations could provide much detailed, accurate information about current condition.

This paper is organized as follows. In section 2, the classic Dempster-Shafer Theory of Evidence is introduced, and the problem of application DST to online diagnosis for operation condition monitoring and failure detection and recognition is analyzed. Here we purposed a temporal weighted evidence combination method together with the procedure of application. In section 3, an example is discussed to illustrate how the temporal weight D-S rule combination method can be applied to online failure identification. Also we compared the result of classic D-S rules of combination with our method.

Xiaoyun WANG et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. METHODOLOGY

2.1. D-S evidence theory

The Dempster-Shafer Theory (DST) or D-S theory of evidence was first introduced at 1960s(Dempster, 1968)(Shafer, 1976). The DST is basically an extension of traditional probabilistic modeling of uncertainty. Currently, the D-S theory of evidence was applied widely in fault diagnosis and recognize for its effectiveness to incomplete, inaccurate or conflict data.

According to the classic D-S theory of evidence, the elements needed to model the problem could be summarized as following:

A frame of discernment Ω, which should be a finite set of all of the possible hypotheses that are mutually exclusive;

A mapping of $m: 2^{\Omega} \to [0,1]$, which defines the basic probability assignment(BPA) of each subset $A \subset \Omega$ of hypotheses and satisfying $m(\varnothing) = 0; \sum_{A \subseteq \Omega} m(A) = 1$. The BPA represents a certain piece of evidence.

A rule of D-S evidence combination, which could be used to yield a new BPA from two independent evidences and their BPAs. There are a number of possible combination rules in application (Sentz, 2002). One of them is the Dempster's Rule, that could be defined as follows

$$m(A) = m_1 \oplus m_2(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - K} \quad (1)$$

$$K = \sum_{B \cap C = \varnothing} m_1(B) m_2(C) \quad (2)$$

.

B and C are subset of hypothesis. K reflects the conflict between B and C, while the higher the K, the greater the conflict between the evidences. It was proven that the Dempster rule of combination meets the commutative and associative laws, which could be depicted as such:

$$(m_1 \oplus m_2) \oplus m_3 = m_1 \oplus (m_2 \oplus m_3)$$

and

$$m_1 \oplus m_2 = m_2 \oplus m_1 .$$

Therefore evidences are treated as equal, as well as the order of evidences dose not affect the result of evidence combination.

2.2. Temporal Weighted Evidence Combination

D-S rule of combination treat evidences equally from different sensor. However, that assumption generally cannot hold during an online condition monitoring. System data and evidence unveiled gradually, sequentially, as time lapsing. What we have identified is only a fraction of the facts. At the early stage of a fault or failure, the symptom could be dim and weak. As the system operation went on, the system performance appears variation, while some symptom may change as well, others could be expired or not valid any more. The creditability of past evidence is not static. Instead it should change in course of timeliness. Evidence that is up-to-date should be assessed as a strong sample. The more recent observations could provide much detailed, accurate information about current condition. At the same time, those past, obsolete evidences only have partial utility, appeared a weak correlation to current faults and errors(Garvin, 1988).

Based on the weighted view of evidence (Yu, 2005), we purposed a temporal weighted combination rules to solve the problems of timeliness of evidence. The weight of each evidences are based on their timestamp properties. The temporal weighted rule combination is:

$$m_{1,2}(A) = \frac{\sum_{B \cap C = A} m_{t1}(B)^{w1} \cdot m_{t2}(C)^{w2}}{\sum_{B \cap C \neq \varnothing} m_{t1}(B)^{w1} \cdot m_{t2}(C)^{w2}} \quad (3)$$

where w_1, w_2 is the temporal weights of time point t_1, t_2 for evidence B and C:

$$w_i = \exp(K(t_i - T)) \quad (4)$$

in which T is the current time (system time). K is a user predefined constant, $K \geq 0$.

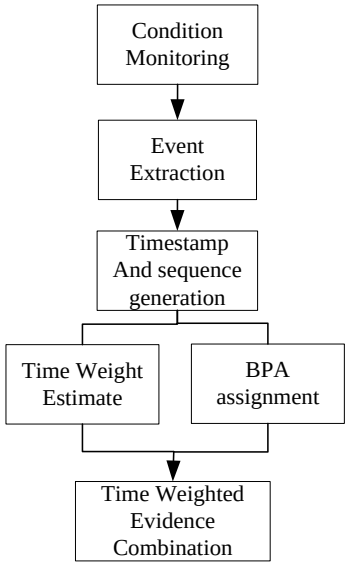


Figure 1. Schematic of the method for time weighted evidence combination.

From the equation (4), we could find some feature of w_i:

a) The temporal weight of the latest evidence is greater than that of those previously evidences.

b) The older of the evidence, the less timeliness and values of the belief, as well as its temporal weight.

c) Evidences with time proximity have similar temporal weights.

d)Temporal weight of on-going evidence has approximate value to 1, which represent it is the most up-to-date evidence.

The workflow of temporal weighted D-S evidence combination method is described as Figure 1. Observed events were extracted from multiple time point to form a temporal evidence sequence. Basic probability assignment is altered by temporal weights in accordance with the time proximity between the observed events and current time. The temporal weights setup is in accordance with its occurring time point. Lately observed evidence could have better influence and support to the hypothesis than those older evidences. Evidences with same timestamps should be allocated with the same temporal weights.

Considering the introduction of temporal weighted combination rules, the combination of multiple evidence is no longer commutative and equally treated, that means each time point we need to recalculate the set of temporal weight w_i, as shown in Figure 2.

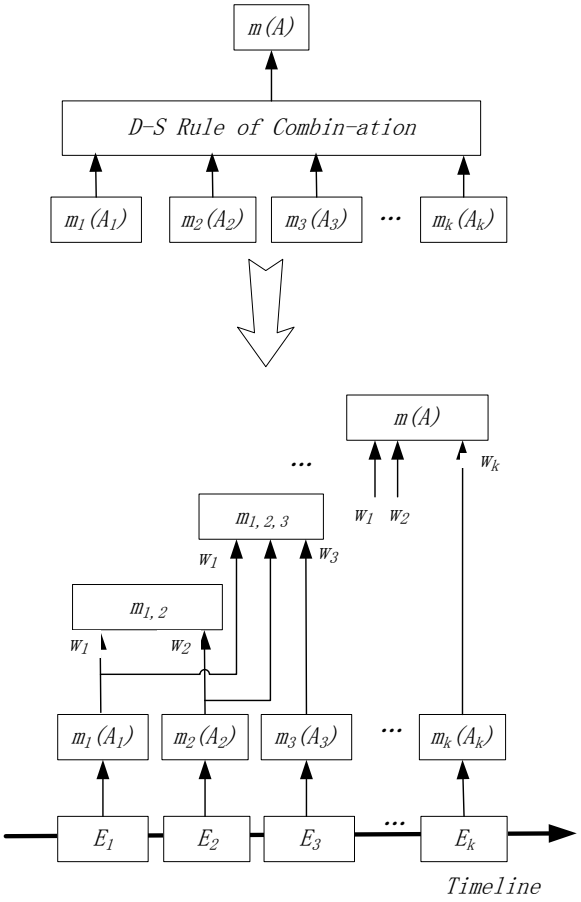


Figure 2. Time weighted D-S evidence combination.

However, this approach might be faced with time complexity for the calculation of w_i at each time point. To simplified the framework, we merged the past combination result into a new evidence at each time point, as shown in Figure 3. The improved framework has better time performance while yield approximately result as Figure 2.

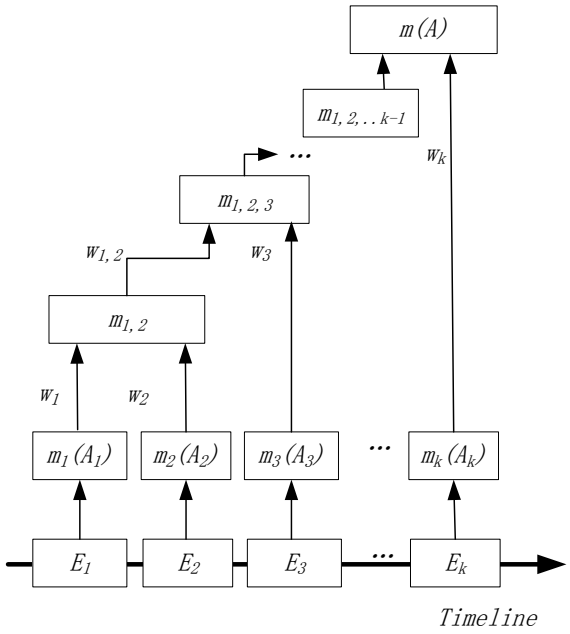


Figure 3. A improved framework of time weighted D-S evidence combination for multiple symptoms.

3. CASE STUDY

In this case, we adopted the dataset of a power generator (Ray, 2007) as an example to illustrate the temporal weight, D-S rule based assessment framework. During its operation, working condition and performance events was monitored periodically.

We need to assess the on-going events and symptoms to identify the type of possible failure(s) in a near real time manner. The challenge lies in that the observing events were ever changing and added in, while the early events and their information might expired, the latest evidence need be combined into the frame to improving the accuracy of results.

3.1. Dataset Preparations

There are three kinds of power generator failures given by domain specialist, namely, h_1, h_2, h_3. The corresponding frame of discernment could be given with

$$\Theta=\{h_1, h_2, h_3, \theta\}$$

where θ is the unknown type of failures.

Event E_1, E_2 and E_3 were reported by system monitoring function, with the corresponding timestamps as $t_1=1, t_2=2, t_3=3$. So we have a sequence of symptom as $\{<E_1,1>,<E_2,2>,<E_3,3>\}$.

To simplify the calculation, we choose the time-weighted constant $K=\ln 2$. As a result, the time weight turns into

$$w_i = \exp(K(t_i - T)) = 2^{(t_i - T)} \quad (5)$$

The BPAs for the hypothesizes of evidence E_1, E_2 and E_3 were given.

Event E_1 was the symptom to failure types h_1, h_2, h_3 with a BPA of 0.7, so:

$$m_{t_1}\{h_1, h_2, h_3\} = 0.7 \text{ And } m_{t_1}\{\theta\} = 0.3$$

E_2 was the symptom of failure h_1 with the belief of 0.9:

$$m_{t_2}\{h_1\} = 0.9 \text{ And } m_{t_2}\{\theta\} = 0.1$$

For event E_3 which was a evidence for failures of h_2, h_3, the BPA is 0.8, so that:

$$m_{t_3}\{h_2, h_3\} = 0.8 \text{ and } m_{t_3}\{\theta\} = 0.2$$

3.2. Temporal Combination of Evidence Sequence

According to equation (5), the time weight could be given for sequence $\{<E_1,1>,<E_2,2>,<E_3,3>\}$.

Event E_1 was detected at $t_1 = 1$. With the new event E_2 was detected at $t_2 = 2$, evidence E_1 and evidence E_2 need to be fused. Table 1 shows the combination rules for m_{t_1,t_2}:

Table 1. Combination of E_1 and E_2

$m_{t_1}(B)^{w_1} \cdot m_{t_2}(C)^{w_2}$	$\{h_1\}$	$\{\theta\}$
$w_1 = 0.5$ $w_2 = 1$	$m_{t_2}\{h_1\} = 0.9$	$m_{t_2}\{\theta\} = 0.1$
$\{h_1, h_2, h_3\}$	$\{h_1\}$	$\{h_1, h_2, h_3\}$
$m_{t_1}\{h_1, h_2, h_3\} = 0.7$	0.753	0.084
$\{\theta\}$	$\{h_1\}$	$\{\theta\}$
$m_{t_1}\{\theta\} = 0.3$	0.493	0.055

According to equation (3),

$$m_{t_1,t_2}\{h_1\} = \frac{\sum_{B\cap C=h_1} m_{t1}(B)^{\frac{1}{2}} \cdot m_{t2}(C)^1}{\sum_{B\cap C\neq\varnothing} m_{t1}(B)^{\frac{1}{2}} \cdot m_{t2}(C)^1} = 0.9$$

$$m_{t_1,t_2}\{h_1, h_2, h_3\} = 0.06$$

$$m_{t_1,t_2}\{\theta\} = 0.04$$

With the event E_3 was detected at time $t_3 = 3$, new evidence added in and the result reflect the influence of up-to-date information. Table 2 shows the combination rules for m_{t_1,t_2,t_3}:

Table 2. Combination of E_1, E_2 and E_3

$m_{t_1,t_2}(B)^{w_{1,2}} \cdot m_{t_3}(C)^{w_3}$	$\{h_2, h_3\}$	$\{\theta\}$
$w_{1,2} = 0.5$ $w_3 = 1$	$m_{t_3}\{h_2, h_3\} = 0.8$	$m_{t_3}\{\theta\} = 0.2$
$\{h_1\}$	$\varnothing$	$\{h_1\}$
$m_{t_1,t_2}\{h_1\} = 0.9$	0	0.19
$\{h_1, h_2, h_3\}$	$\{h_2, h_3\}$	$\{h_1, h_2, h_3\}$
$m_{t_1,t_2}\{h_1, h_2, h_3\}$ $= 0.06$	0.196	0.049
$\{\theta\}$	$\{h_2, h_3\}$	$\{\theta\}$
$m_{t_1,t_2}\{\theta\} = 0.04$	0.16	0.04

The combination at time $t_3 = 3$ as shown:

$$m_{t_1,t_2,t_3}\{h_1\} = 0.299$$

$$m_{t_1,t_2,t_3}\{h_2, h_3\} = 0.560$$

$$m_{t_1,t_2,t_3}\{h_1, h_2, h_3\} = 0.077$$

$$m_{t_1,t_2,t_3}\{\theta\} = 0.063$$

Here we had combine the sequence $\{<E_1,1>,<E_2,2>,<E_3,3>\}$ at $t_3 = 3$.

In Table 3 we compared the results of classic D-S approach and our temporal weighted combination method. Apparently, from row $t_3 = 3$ we can see that the temporal weighted approach is more sensitive to latest, up-to-date evidence, which yield a higher belief for hypothesis set $\{h_2, h_3\}$ in favor of the newly observed evidence $<E_3,3>$. Also we could infer from the line $t_2 = 2$ that when the latest evidence was similar to the former ones, the output beliefs of temporal weighted combination method is only slightly different to classic D-S approach.

Table 3. Comparison of temporal weighted combination method and classic D-S evidence combination

Time	BPA	Classic D-S	Temporal Weighted
$t_2 = 2$	$m_{t_1,t_2}\{h_1\}$	0.9	0.9
	$m_{t_1,t_2}\{h_1, h_2, h_3\}$	0.07	0.06
	$m_{t_1,t_2}\{\theta\}$	0.03	0.04
$t_3 = 3$	$m_{t_1,t_2,t_3}\{h_1\}$	0.511	0.299
	$m_{t_1,t_2,t_3}\{h_2, h_3\}$	0.227	0.560
	$m_{t_1,t_2,t_3}\{h_1, h_2, h_3\}$	0.034	0.077
	$m_{t_1,t_2,t_3}\{\theta\}$	0.227	0.063

4. CONCLUSION

The Dempster-Shafer Theory of evidence based model has been widely used to multi sensor fault detection and recognition. As an improvement to the DST, the temporal weighted evidence combination method could be beneficial to the balance of long term trend and abrupt fault recognition, especially for the online health management applications, compared with the classic DST combination method.

Our contribution could be summarized as follows: First, the problem of obsolete evidence of real time monitoring and diagnosis is analyzed. Then the temporal weighted evidence combination method is purposed. To make the method for efficiency, an improved framework that accumulates the past combination result is suggested. Furthermore, a case study was discussed to illustrate the temporal weighted D-S rule based assessment framework.

ACKNOWLEDGEMENTS

Funding for this paper was provided by BeiHang University. The authors gratefully acknowledge the sponsor of the School of Reliability and System Engineering, BeiHang University.

REFERENCES

Yang, B.S., Kim, K. J., (2006). Application of Dempster-Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing*, 20:403–420.

Parikh,C.R., Pont, M.J., Jones, N.B., (2001). Application of Dempster-Shafer theory in condition monitoring systems: A case study, *Pattern Recognition Letters*, 22 (6-7): 777-785.

Fang, L., Wang, C., et.al, (2010). A Framework for Network Security Situation Awareness Based on Knowledge. *2nd International Conference on Computer Engineering and Technology*, Chengdu, China

Zomlot, L., Sundaramurthy. S., (2011). Prioritizing Intrusion Analysis Using Dempster-Shafer Theory, *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, Chicago, Illinois, USA

McKeever, S., (2009). *Recognising Situations Using Extended Dempster-Shafer Theory*, Doctoral dissertation. National University of Ireland, Dublin

Beranek, L., Knizek, J., (2013).The Use of Contextual Information to Detection of Fraud on On-line Auctions. *Journal of Internet Banking and Commerce*, vol. 18, no.3

Dempster, A., (1968). A Generalization of Bayesian inference. *Journal of the Royal Statistical Society*, pp. 205–247.

Shafer, G., (1976). *A Mathematical Theory of Evidence*, New Jersey, Princeton University Press.

Sentz, K., (2002). Combination of Evidence in Dempster-Shafer Theory, Binghamton University, Binghamton, NY

Garvin, D., (1988). Managing quality. NY: Free Press.

Yu, D,. Frincke, D., (2005). Alert Confidence Fusion in Intrusion Detection Systems with Extended Dempster-Shafer Theory, *43rd ACM Southeast Conference*, Kennesaw, GA

Kay, R.U., (2007). Fundamentals of the Dempster-Shafer theory and its applications to system safety and reliability modeling. *Reliability: Theory and Applications*. vol 2(3-4), pp. 173-185

BIOGRAPHIES

Xiaoyun Wang Phd candidate of Aerospace Engineering, School of Reliability & System Engineering, BeiHang University, Beijing, China, telephone: (8610) 82317665, email:wangxys@gmail.com

Tingdi Zhao Professor of Engineering, School of Reliability & System Engineering, BeiHang University, Beijing, China, telephone:(8610)82316570, email:ztd@buaa.edu.cn

Modeling Hydraulic Components for Automated FMEA of a Braking System

Peter Struss, Alessandro Fraracci

Tech. Univ. of Munich, 85748 Garching, Germany
struss@in.tum.de

ABSTRACT

This paper presents work on model-based automation of failure-modes-and-effects analysis (FMEA) applied to the hydraulic part of a vehicle braking system. We describe the FMEA task and the application problem and outline the foundations for automating the task based on a (compositional) system model. Models of the essential hydraulic components suitable to generate the predictions needed for the FMEA are introduced and the required models of the control software outlined. These models are based on constraints, rather than simulation, and capture the dynamic response of the systems to an initial situation based on one global integration step and determine deviations from nominal functionality of the device. We also present the FMEA results based on this model.

1. INTRODUCTION

Failure-modes-and-effects Analysis (FMEA) is performed by groups of experts during the design phase of a system. Its core is to exhaustively go over all potential component faults and predict their impact on the functionality of the system in order to assess whether they can lead to a critical situation and violate safety requirements, and take steps to minimize or mitigate the negative impact through a design correction.

FMEA was originally developed in the military area (MIL, 1980) and has become a mandatory task in the aeronautics and automotive industries (see e.g. (SAE, 1993)), meanwhile as part of international standards for functional safety (e.g. ISO 26262 in the automotive industries, (ISO 2011)) and receives increasing interest in other areas, such as automation systems.

The main result of the analysis is a table that relates certain scenarios (such as "Braking in forward motion"), components or subsystems and their faults ("valve stuck open") to the effects caused by them in the respective scenario, possibly at component level, next level, and system level, ("right front wheel overbraked; vehicle yawing to the right") and some other assessments (e.g. criticality, detectability, suggested design changes).

The analysis is performed by groups of experts, consuming precious time and labor, and repetitive, because it has to be redone or revised for each variant or version of a system and each revision of a design. Current computer support to reduce the effort and time is fairly poor and mainly limited to editors and data handling. The key part of the analysis, inferring the effects of the assumed faults, remains the task of the human experts. Although a major part of this analysis is not very sophisticated, but rather routine and mechanistic, it requires knowledge about the involved components and reasoning about the (physical and software) system. Hence, computer systems substantially supporting it have to be **knowledge-based systems**. More specifically:

- a **model-based** solution is required that can reason about how the (mis-)behavior of components and their interaction establishes the (mis-)behavior of the overall system, because, during early design stages, only a blueprint may be available. (Even when a physical prototype exists, it may be too costly, risky, or even impossible to implant certain faults in the physical system.)
- Exact parameter values of the design may still be undetermined. Hence, the analysis cannot be based on numerical, but only on **qualitative** models.
- Even if the parameters do have fixed numerical values, the analysis is inherently **qualitative** both w.r.t input (classes of faults, such as "a leakage", rather than "leakage of size x") and relevant effects ("loss of pressure in wheel brake" and "potentially reduced deceleration").

For both reasons, numerical models (e.g. Matlab/Simulink, Modelica models) are useless and could, at best, produce some incomplete hints, based on sampling an infinite space

Peter Struss et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of space of scenarios and faults. In fact, we are not aware of any serious attempt of using numerical models for this purpose in practice.

- The modeling effort must be low to handle a class of systems and to support repetitive FMEA of design variants and modifications. This needs to be addressed by **compositional modeling**, which has to be based on a library of generic, context-independent component models.

The systems that offer support to the automated generation of fault-effect associations in the context of FMEA are based on qualitative models. The AutoSteve system (Price, 2000) was specialized on performing FMEA of **electrical** car subsystems. The AUTAS project developed a **generic** FMEA tool with applications to electrical, hydraulic, pneumatic, and mechanical systems in aeronautic systems (Picardi *et al.*, 2004).

In collaboration with a German car manufacturer, we applied this algorithm to FMEA of a novel braking system, which confronted us with the need for models of hydraulic components, especially valves, that are, on the one hand, general enough to be reusable and, on the other hand, powerful enough to deliver the predictions relevant to FMEA of braking systems.

In this paper, we present the core of models that have proven to successfully produce the results needed for FMEA of the braking system. The key features of the models are that they

- capture one integration step, but avoid any simulation and are stated in terms of constraints (finite relations),
- are compositional and context-independent,
- analyze how a stimulus in terms of a local pressure change (e.g. pushing a brake pedal) propagates through the system,
- capture qualitative deviations of pressure and flow from their nominal values resulting from component faults,
- can be complemented by models of the control software functions for both their correct and their faulty behavior, due to the high level of abstraction.

The focus of the work reported in this paper is on automatically determining the local and global effects of each failure mode (i.e. component fault). It first describes the case study, FMEA of braking systems, and then summarizes the foundations of model-based FMEA. In section 4, we present the key parts of the models. The results obtained for FMEA are discussed in section 5. Section 6 briefly outlines foundations for modeling the embedded software.

2. THE BRAKING SYSTEM

The target is a novel braking system whose details are proprietary. For safety reasons, it still has to comprise the traditional braking function. Therefore, we use this part of the system in order to illustrate our solution.

A standard braking system is mainly composed of hydraulic and mechanical components and the electronic control unit (ECU) and its software. It contains a tandem pedal actuation unit (with two pistons and two chambers), valves (inlet and outlet types) and wheel brakes, shown in Figure 1.

The pedal actuation block (top right) comprises two pistons (PA_P1 and PA_P2) and the two chambers (PA_C1 and PA_C2), where PA_P1 is directly affected by pushing the brake pedal. Each chamber produces pressure for one diagonal wheel pair, and each wheel brake (WB11, 12, 21, 22) sits between an inlet valve and an outlet valve.

The inlet-valves (M_VI11, 12, 21, 22) are piloted check valves; during standard braking (i.e. with no command), they are open, while the outlet-valves (M_VO11, 12, 21, 22) are closed. Pushing the brake pedal causes pressure to build

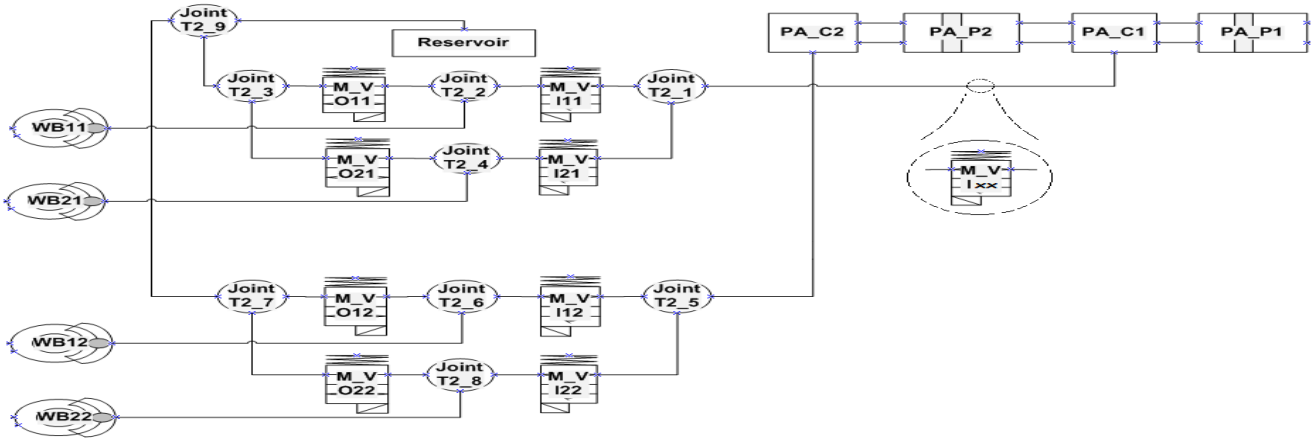


Figure 1. Braking system. Pressure is generated by two pistons, PA_P1,2, in two chambers, PA_CA1,2, and reaches the wheel brakes, WBij, via open inlet valves, M_VIij, while outflow is blocked by closed outlet valves, M_VOij. The impact of inserting another valve, M_Vixx, is discussed in section 5.3

up in the wheel brakes. Inlet valves always allow a flow back from the wheel brakes, which causes the diminishing of the wheel brake pressure if the brake pedal is released.

When operated under the Anti-lock-braking system (ABS), the valves are controlled by commands from the ECU. The pressure-build-up phase is the scenario described above. For pressure maintenance, the inlet valve is closed. If the speed sensors indicate that the wheels tend to lock up, the outlet valves are opened to release pressure, let the wheels spin again and, thus, enable steering of the vehicle. Then the cycle is entered again.

Typical inferences required for FMEA are

- If an inlet valve is stuck closed under normal braking, the respective wheel will be underbraked (reduced deceleration).
- The same holds if an outlet valve is stuck open under normal braking.
- If an outlet valve is stuck closed during the pressure release phase of ABS braking, the respective wheel will be overbraked, because the pressure is not released.
- An inlet valve being stuck open during this phase will have the same impact.

Other faults are leakages of the wheel brakes and the chambers, the wheel brakes and pistons being stuck, bad sensors etc. Also bugs in the embedded software have to be considered, which becomes an increasingly important aspect in functional safety.

3. MODEL-BASED FMEA

Predicting the principled impact of (classes of) faults in (classes of) scenarios is the core of the FMEA task. In this section, we summarize the logical foundation of model-based FMEA, which have been developed in the AUTAS project (see (Picardi et al., 2004), (Fraracci, 2009)), implemented as an inference engine in Raz'r (OCC'M, 2014), and applied to various aircraft subsystems.

3.1. Relational Models

As motivated in the introduction, models supporting FMEA have to be qualitative. We use finite qualitative relations over variables. Hence, a behavior model is regarded as a relation R over a set of variables that characterize a component or system: $R \subset DOM(\underline{v})$, where $\underline{v}$ is a vector of system variables with the domain $DOM(\underline{v})$, which is the Cartesian product

$$DOM(\underline{v}) = DOM(v_1) \times DOM(v_2) \times ... \times DOM(v_n).$$

So, a relation R (i.e. a *constraint*) is a subset of the possible behavior space.

If elementary model fragments R_{ij} are related to behavior modes $mode_i(C_j)$ of the component C_j, then an aggregate system (under correct or faulty conditions) is defined by a mode assignment $MA = \{mode_i(C_j)\}$ which specifies a unique behavior mode for each component of this aggregate whose model is obtained as the join of the mode models, i.e. the result of applying a (complete version of) constraint satisfaction to $\{R_{ij}\}$:

$$R_{MA} = \bowtie R_{ij}.$$

3.2. Formalization of FMEA

To support FMEA, it is necessary to determine whether the effects of a certain component fault (represented as a mode assignment *MA*) violate an intended function of the system. If the function is considered as part of *GOALS*, then the task might mean to check whether the fault model FM_{MA} is inconsistent with the function:

$$FM_{MA} \cup GOALS \vdash^{?} \bot$$

Often, the analysis is carried out for particular mission phases (such and "cruising" or "landing" of an aircraft) or scenario S_k (e.g. the three phases of the ABS braking as explained above):

$$FM_{MA} \cup S_k \cup GOALS \vdash^{?} \bot$$

In practice, FMEA is not carried out this way, but by specifying effects E_i, which are specific violations of the intended function (*GOALS*), for instance too high and too low deceleration of a wheel, i.e. underbraking and overbraking:

$$S_k \cup E_i \vdash \neg GOALS \ ,$$

and the analysis determines the effects that may occur under a particular failure mode:

$$FM_{MA} \cup S_k \cup E_i \vdash \bot$$

Since models, scenarios, and effects can all be represented by relations, we can characterize and compute the effects of the FM_{MA} as follows:

- $R_{MA} \bowtie S_k \subset E_1$
  if the failure mode is included in effect, then the effect will **definitely occur** (case E_1 in Figure 2)
- $R_{MA} \bowtie S_k \cap E_2 = \varnothing$
  if the intersection is empty, the effect **does not occur** (case E_2)
- otherwise
  the effect **may occur**: E_3

The above checks can be performed using general techniques, such as constraint solvers (Rossi et al., 2008) or logical reasoning engines that can determine consistency

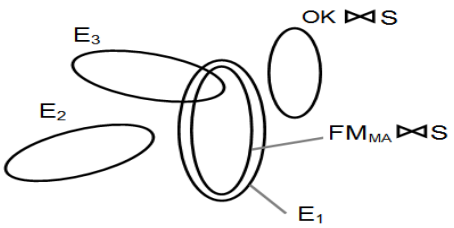


Figure 2. Determining effects

and entailment. We use the FMEA engine of Raz'r mentioned above (OCC'M, 2014).

3.3. Deviations Models – Formalization

FMEA is about inferring deviations from nominal system function due to a deviation from nominal component behavior. Hence, not the magnitude of certain quantities matter, but the fact whether or not they deviate from what is expected under normal or safe behavior.

This is why deviation models (Struss, 2004) offer the basis for a solution: they express constraints on the deviations of system variables and parameters from the nominal behavior and capture how they are propagated through the system.

For each system variable and parameter v_i, the deviation is defined as the sign of the difference between the actual and a reference value:

$$\Delta v := \text{sign}(v_{act} - v_{ref}).$$

Then algebraic expressions in an equation can be transformed to deviation models according to rules like

$$a + b = c \Rightarrow \Delta a + \Delta b = \Delta c$$
$$a * b = c \Rightarrow a_{act} * \Delta b + b_{act} * \Delta a - \Delta a * \Delta b = \Delta c ,$$

where +, -, * on the right-hand side should be interpreted as operators over the sign domain.

Furthermore, for any monotonically growing (section of a) function $y = f(x)$, we obtain $\Delta y = \Delta x$ as an element of a qualitative deviation model.

For instance, the deviation model of a valve is given by the constraint

$$\Delta Q = A * (\Delta P_1 - \Delta P_2) + \Delta A * (P_1 - P_2) - \Delta A * (\Delta P_1 - \Delta P_2)$$

on the signs of the deviations of pressure (ΔP_i), flow (ΔQ), and area (ΔA). This constraint allows, for instance, to infer that P_1 being too large ($\Delta P_1 = +$) causes an increased flow ($\Delta Q = +$), if P_2 and the area remain unchanged ($\Delta P_2 = 0$, $\Delta A = 0$) and the valve is not closed ($A = +$). Such qualitative deviation models specify finite relations over the qualitative variables and can be constructed from first principles (differential) equation models, if they exist. Under certain conditions (piecewise monotonic functions) these relations can be calculated automatically from numerical models (Struss et al., 2011).

Note that in contrast to model-based diagnosis, where we may use the very same models, we do not face the problem of determining whether a certain sensor value indicates a qualitative deviation or not: in FMEA, there are no measurements; a deviation is simply **assumed** as the starting point of the analysis.

4. HYDRAULIC MODELS

The literature on qualitative modeling does not deliver a ready-made library of hydraulic models that could be used for real applications like the one we are tackling. Especially for valves, most of the proposed models compile strong assumptions about the context into the models, which makes them inappropriate for a library of generic, reusable component models. What is it that makes hydraulic modeling hard? While we can easily model, for instance, a resistor network by simultaneous equations characterizing the steady state, the analysis of hydraulic systems often focuses on the transitions, and the finally reached equilibrium may be uninteresting (e.g. all connected parts with equal pressure). Pressures determine flows, which in turn determine change of pressure. Hence, the analysis has to include some integration step (in the mathematical sense), and our component models duplicate variables to describe states "before" and (directly) "after".

Another problem dimension, which is not dealt with in this paper, is related to the fact that often, the nature of the stuff that flows cannot be ignored, e.g. when there is air in a hydraulic circuit.

In the following, we present the core pieces of the qualitative hydraulic model that we used to solve the FMEA task. Our starting point was our early work on modeling for diagnosis of braking systems (Struss *et al.*, 1997), and we created

- a **relational** model that
- **qualitatively** captures the system's direct **response** to some **initial condition**, especially
- in terms of **deviations** from nominal behavior, and
- can be **used by the FMEA engine** whose basis was outlined in section 3.2.

Despite its simplicity, it turns out to be quite powerful and appropriate for generating the kind of information needed for the FMEA task. We first characterize its scope by discussing the most important requirements and modeling assumptions underlying it and then present the various "slices" of the key component models, namely valve and volume.

4.1. Modeling Methodology and Assumptions

In the current model, we assume that there is one source of pressure, or, more precisely, a unique maximal pressure level generated by components or some external force. In our application, this is determined by the driver pushing the brake pedal. It is not fixed to a particular numerical value, but, rather, by the fact that the pressure in the system cannot exceed it. We are convinced that the approach can be extended to multiple source levels, but did not implement such a model and make no claims.

This assumption is reflected by the chosen domain for pressure:

$$\text{PosSign3}:=\{0, (+), +\},$$

where + is the source pressure (and maximal), 0 corresponds to the sink (in our case the reservoir of the liquid), and (+) is any pressure in between. For pressure drops and flows, only their direction matters, i.e. their domain is Sign = {-, 0, +}. Valves are assumed to be either closed (A = 0) or open (A = +), which does not imply they are **completely** open.

The next assumption (a requirement of our application) is that the interest is in determining the systems direct response to an initial situation. To illustrate what this means (and what is excluded), consider the right-hand part of Fig. 3 with a volume component Vol_2, with initial pressure 0, connected via open valves on the right to a volume Vol_1 with pressure P=+ in the initial scenario S_0, and on the left to another volume Vol_3 with initial pressure (+). The state following this initial situation will be a state with positive inflows Q into Vol_2, and this is what the model should predict (scenario S_1 in Fig. 3). There may be a future state, in which the pressure in Vol_2 exceeds the one in Vol_3, and the flow through the respective valve reverses. This is not what we are interested in, and accordingly, we exclude such multiple changes of qualitative values. Also, no other event should occur during the period of interest, especially no valve changes its state. We furthermore assume pressure to be homogeneous in a volume and ignore time required to achieve or approximate the situation.

To simplify the presentation in this paper, we assume that there are no deviations in the initial situation. This assumption can be dropped if the system response to a deviating initial situation is of interest.

The modeling is not ad-hoc, but follows a **clear and general methodology** that can be applied to other components and systems. A qualitative deviation component model is constructed from an equation-based model M_e as the union of five sets of constraints, three obtained as **transformations of M_e**:

- $Q(M_e)$: the qualitative abstraction of M_e

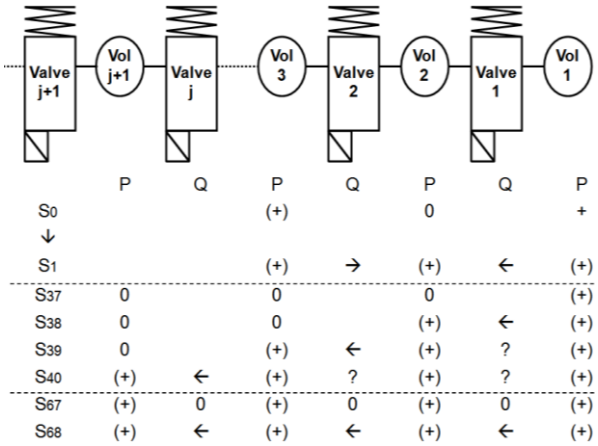


Figure 3. Volume-Valve sequence

- $\partial(M_e)$:the qualitative abstraction of the derivative version of M_e
- $\Delta(M_e)$: the qualitative deviation model of M_e

and two set of constraints representing the **qualitative integration constraints**, which are generic and **independent of M_e**:

- I(x): the qualitative integration constraint for the variables
- I(Δx): the qualitative integration constraint for the deviations.

	Valve	Volume
Base model $Q(M_e)$	$T_1.Q=A*(T_1.P-T_2.P)$ $T_1.Q = -T_2.Q$	$T_1.Q = \partial P$
Base model derivative $\partial(M_e)$	$T_1.\partial Q = A*(T_1.\partial P-T_2.\partial P)$ $T_1.\partial Q = -T_2.\partial Q$	
Deviation model $\Delta(M_e)$	$T_1.\Delta Q = \Delta A*P_{diff} + +A*\Delta P_{diff}-\Delta A*\Delta P_{diff}$ $P_{diff} = T_1.P-T_2.P$ $T_1.\Delta Q = -T_2.\Delta Q$	$T_1.\Delta Q = \Delta\partial P$
Continuity Integration Persistence I(x)	Q_0 | ∂Q | Q - | * | - 0 | - | - 0 | 0 | 0 0 | + | + + | * | +	P_0 | ∂P | P 0 | 0 | 0 0 | + | (+) (+) | * | (+) + | - | (+) + | 0 | +
Integration Deviation I(Δx)	$T_i.\Delta\partial Q = T_i.\Delta Q$	$\Delta P = \Delta\partial P$

Figure 4. The elements of valve and volume models

We present the different elements of the models, which are summarized in Figure 4. We do so step by step in order to demonstrate the necessity of each model slice and its contribution.

4.2. Base Models

The core of the models is given by the qualitative abstractions of the standard (differential) equations. A key requirement is that the component models are local and context-independent in order to be compositional as required by the application task.

For the **valve**, the terminals T_i are its hydraulic connections (it has another one for the control command). With the convention that a positive flow is going into the respective component, we obtain

$$T_1.Q = A^* (T_1.P\text{-}T_2.P) ,$$

where pressure subtraction

$$\text{-} : \{0, (+), +\}\times\{0, (+), +\} \rightarrow \{\text{-}, 0, +\}$$

is defined as

$$0 - 0 = + - + = 0,$$
$$+ - (+) = + - 0 = (+) - 0 = +$$
$$0 - (+) = 0 - + = (+) - + = -$$
$$(+) - (+) \text{ unrestricted.}$$

The second element is Kirchhoff's Law (see Fig. 4, row 1).

Since A is the **actual** opening of the valve, these elements apply to all behavior modes of a valve except leakages.

The base model of a **volume** is straightforward. To simplify the presentation, we consider a volume with only one terminal (like the wheel brake). If there is more than one terminal, $T_1.Q$ is replaced by the sum of all flows across all terminals (or the volume is connected to a joint capturing the various flows, as done in the brake model). In case of a leakage, also the resulting leak flow has to be included. ∂P denotes the qualitative derivative with the domain Sign.

The results obtained by this base model do not always contain an answer relevant to the FMEA task. In our brake system, normal braking happens when the inlet valve is open and the outlet valve is closed. The consequence is pressure (+) in the wheel brake. If the outlet valve is stuck-open, there will be an outflow (after one integration step). The wheel brake pressure is still (+). But the important point is: it is less than under nominal conditions. Therefore, we add a layer of deviation models, as shown in Figure 4.

4.3. Deviation Models

The deviation models are easily obtained from the algebraic equations of the base models. However, they are quite powerful and provide the prediction we need for FMEA in the scenario discussed above: the inflow via the inlet valve will have a deviation 0, while the flow towards the outlet valve has a negative deviation (being negative instead of 0), and, hence, will cause a negative deviation $\Delta\partial P$ ("reduced pressure built-up").

Again, the deviation model applies to each instance of time. But still, we need to answer the question how we represent and predict the overall system response properly.

4.4. Integration, Continuity, Persistence

This model, which applies to every point in time, still has limited utility. Consider again a sequence of three or more connected volumes (as in Figure 3), each with initial pressure 0, except for Vol_1, which has a pressure (+). What we would like to predict is a flow through all valves from right to left (scenario S_{37} in Fig. 3). The model as it stands will predict a flow into Vol_2 and zero flows, otherwise (S_{38}). Of course, the pressure derivative in Vol_2 is positive. Hence, after integration, the pressure becomes (+), too, and applying the model will lead to a flow from Vol_2 to Vol_3 – but leave the flow from Vol_1 to the second Vol_2 unrestricted, because of pressure=(+) for both (S_{39}). If there are n more volumes, n integration steps are required in order to let the flow reach the last one – and leave all other flows undetermined. – Obviously, this is not what we need.

In our model, we consider two temporal slices of the system behavior: the initial situation and the one capturing the direct global system response, i.e. a representation of the state after the effect of pressure differences has been propagated to all (connected) parts of the system. This means, we neglect the time needed for this propagation and apply some kind of "temporal factorization" (Pietersma & van Gemund, 2007).

The initial state is characterized by variables P_0, Q_0, etc., while the next state is represented by P, Q, etc.

Then the integration step can be represented as a constraint on different variables, namely P_0, ∂P, P. The crucial point is that we do **not** choose ∂P_0, but ∂P, i.e. the derivative **after** the impact. Figure 4 shows the respective constraint in row 4, column 3. It expresses more than the continuous transition from P_0 to P dependent on ∂P. It excludes transitions from (+) to + or 0, expressing the restriction of the predictions to the next state (which implies the exclusion of state-changing events).

Starting from some initial situation and the respective values of P_0, Q_0, etc., how can we determine ∂P instead of only ∂P_0? This is supported by the constraint on flows shown in row 4, column 2 of Figure 4. Again, it captures more than continuity: non-zero flows are considered to be persistent, which again expresses the restriction to the next qualitative state and the exclusion of events that change the direction of flow. This achieves the intended prediction, for instance, for the volume sequence discussed above: Q_0 and hence, also Q from Vol_1 to Vol_2 is determined to be non-zero, which

suffices to determine ∂P = + and P = (+) for Vol_2. This implies a positive flow into Vol_3, etc.

Without further distinctions between sink and source pressures, i.e. within (+), the model developed, so far, may appear quite weak, being unable to determine the direction of flow between two volumes with pressure (+). Consider another initial scenario, S_{67}, for the hydraulic chain in Fig. 3, where initially, all volumes have pressure (+), the valves are open, but there are no flows across them (because all volumes have exactly the same pressure). If we connect Vol_1 to a source (pressure +) and the left-most valve to a sink (pressure 0), again we expect a flow from right to left (S_{68}). However, the model slices presented, so far, are unable to derive this, because the inflow to Vol_1 leaves its pressure at (+), and the flow through $Valve_1$ remains undetermined. What enables us to predict the change is the consideration that the pressure in Vol_1 has increased, exceeds the one in Vol_2 and, hence, produces a flow into Vol_2, and so on. We can capture this by adding a derivative of the base model that links change in pressure and change in flow, as shown in row 2 of Fig. 4 (We omit producing constraints involving the second derivative, what would happen for the volume). This model successfully generates the expected result S_{68}.

Finally, we add a constraint that integrates the deviations (row 5 of Figure 4). Intuitively, this states that if the derivative of a quantity deviates from the nominal value, then so does the quantity itself. This is based on the assumption that the initial situation does not contain deviations. If it is dropped, an initial pressure deviation has to be added.

5. FMEA RESULTS

5.1. Scenarios

We used the model whose core has been outlined in section 4 to produce an FMEA of the braking system described in section 2 for a number of scenarios: braking and non-braking with/without ABS for a moving/no-moving car. In the following, we focus on the scenario "Standard braking while car moving", which is identical to the 1st phase of ABS braking as explained in section 2. This scenario is defined as:

- no commands to all valves: Cmd = 0 (i.e. under normal conditions inlet valves open, outlet valves closed)
- the initial hydraulic pressure of all wheel-brakes are zero: $WB_{xy}.P_0 = 0$
- velocity $v > 0$ for all: $WB_{xy}.v$ = +
- constant pressure P on the piston PA_P_1 exerted by the brake pedal: $PA_P_1.P$ = +
- no deviation of the pedal pressure: $PA_P_1.\Delta P = 0$ and $PA_P_1.\Delta\partial P = 0$

For the "maintain pressure" phase, the commands to the inlet valves are set to 1, and the wheel brake pressures are (+) (from the previous phase). In the "release pressure" scenario, the commands to the outlet valves also become 1.

5.2. System Level Effects

The system effects are defined by the experts as the relevant deviations from the intended function. For the braking system, this includes the following effects:

- **soft pedal,** P = +; ΔP = 0 and $\Delta\partial pos$ = +; where pos indicates the position of piston PA_P_1: when pushed (without deviation), the piston (and, hence, the pedal) moves faster than normal
- **hard pedal,** like soft pedal with $\Delta\partial pos$ = -
- **underbraking,** reduced deceleration of a wheel:
  $WB_{xy}.\Delta\partial v$ = + where xy indicates the wheel involved
- **overbraking,**
  too much deceleration: $WB_{xy}.\Delta\partial v$ = -
- **potential no steering,** both front wheels are underbraked (and, hence, may lock up)
- **yawing to left**,
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$ + $WB_{22}.\Delta\partial v$-$WB_{12}.\Delta\partial v$ = +
  AND NOT
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$+$WB_{22}.\Delta\partial v$ -$WB_{12}\Delta\partial v$ = -
  where:
  WB_{21}: left front wheel; WB_{11}: right front wheel;
  WB_{22}: left rear wheel; WB_{12}: right rear wheel .
  This means: underbraking of at least one wheel on the right-hand side or overbraking of at least one wheel on the left-hand side and no possibly counteracting under/overbraking.
- **yawing to right**
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$ + $WB_{22}.\Delta\partial v$-$WB_{12}.\Delta\partial v$ = -
  AND NOT
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$+$WB_{22}.\Delta\partial v$ -$WB_{12}\Delta\partial v$ = +
- **potential yawing**
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$ + $WB_{22}.\Delta\partial v$-$WB_{12}.\Delta\partial v$ = -
  $WB_{21}.\Delta\partial v$-$WB_{11}.\Delta\partial v$ + $WB_{22}.\Delta\partial v$-$WB_{12}\Delta\partial v$ = +
  Some over/underbraking, but none of the above cases (i.e. potential compensation of yawing)
- **loss of liquid,** $Qleak_x$ =+, where $Qleak_x$ is the leakage liquid flow and x indicates (as above) the respective wheel involved.

5.3. Results

The qualitative model has been implemented in Raz'r (OCC'M, 2014), an environment for model-based diagnosis, prediction, and FMEA. Partial results for the scenario "Standard braking while car is moving" are shown in Fig. 5.

Scenario	Part	Failure mode	Local effect	System level effect
Braking_CarMoving	PA_C1	SealBroken	>>no local effect<<	:SoftPedal
Braking_CarMoving	PA_C1	AirInChamber	>>no local effect<<	:SoftPedal
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition	HardPedal	
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition	FixedInNotPushedPosition	:WB11_Underbraked
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition		:WB21_Underbraked
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition		:WB12_Underbraked
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition		:WB22_Underbraked
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition		:HardPedal
Braking_CarMoving	PA_P1	StuckInNonBrakingPosition		:PotentialYawing
Braking_CarMoving	PA_P1	StuckInBrakingPosition	HardPedal	:HardPedal
Braking_CarMoving	PA_P2	StuckInNonBrakingPosition	HardPedal	
Braking_CarMoving	PA_P2	StuckInNonBrakingPosition	FixedInNotPushedPosition	:WB12_Underbraked
Braking_CarMoving	PA_P2	StuckInNonBrakingPosition		:WB22_Underbraked
Braking_CarMoving	PA_P2	StuckInNonBrakingPosition		:HardPedal
Braking_CarMoving	PA_P2	StuckInNonBrakingPosition		:PotentialYawing
Braking_CarMoving	PA_P2	StuckInBrakingPosition	HardPedal	:HardPedal
Braking_CarMoving	PA_C2	SealBroken	>>no local effect<<	:SoftPedal
Braking_CarMoving	PA_C2	AirInChamber	>>no local effect<<	:SoftPedal
Braking_CarMoving	M_VI11	BlockedClosed	NoFlow	
Braking_CarMoving	M_VI11	BlockedClosed	ReducedFlow	:WB11_Underbraked
Braking_CarMoving	M_VI11	BlockedClosed		:HardPedal
Braking_CarMoving	M_VI11	BlockedClosed		:YawingToLeft
Braking_CarMoving	M_VI11	BlockedOpen	>>no local effect<<	:>>no system level effects<<
Braking_CarMoving	WB22	Leakage	Underbraked	:WB22_Underbraked
Braking_CarMoving	WB22	Leakage		:SoftPedal
Braking_CarMoving	WB22	Leakage		:WB22_LossOfLiquid
Braking_CarMoving	WB22	Leakage		:YawingToRight
Braking_CarMoving	WB22	StuckInNonBrakingPosition	Underbraked	:WB22_Underbraked
Braking_CarMoving	WB22	StuckInNonBrakingPosition		:YawingToRight
Braking_CarMoving	WB22	StuckInBrakingPosition	>>no local effect<<	:>>no system level effects<<

Figure 5. Partial FMEA (omitting repetitive results)

Columns 2 and 3 refer to the respective component and failure mode, while column 4 states the effects local to this component, and column 5 contains the system level effects. This table, which is generated within seconds (as opposed to person days if carried manually), is complete and correct when compared to FMEA tables produced by experts.

Despite its simplicity, the model turns out to be quite powerful. To illustrate this, consider the table entry for the inlet valve M_VI_{11} BlockedClosed in Figure 5. It predicts that the respective Wheel brake, WB_{11} is underbraked, while WB_{21} behaves normally, because, after all, it receives the proper pressure.

When we insert another valve between the chamber PA_C1 (with pressure +) and JointT2_1 the valve M_IV_{xx} indicated in Fig. 1), then besides WB_{11} underbraked, also WB_{21} overbraked is predicted, because of a higher flow through M_IV_{21} due to the blockage of M_IV_{11}.

6. SOFTWARE MODELS

Including the consideration of the embedded software and, hence, in our approach, a qualitative deviation model of it, is necessary for two reasons:

- the impact of a sensor fault can only be analyzed by considering how the software functions that depend on the sensor value process it to determine actuator signals to the physical components,
- the software itself may contain bugs that lead to behavior deviations of the controlled physical system.

In the following, we briefly outline the basis for modeling the software appropriately and refer to Struss (2013) for the principles and Struss & Dobi (2013) for an application.

In our case study, for investigating the impact of a failure of a sensor that measures the rotational speed of a, we need a model of the intended behavior of the ECU, more precisely the software functions that control the valves based on the measured wheel speed: it has to issue a command, *cmd=1*, when the wheel speed drops below a certain threshold. For two different thresholds, the commands cause an inlet valve to close and an outlet valve to open, respectively. In our context, the only interesting aspect is how the (correct) function propagates a deviation of a sensor value (or a missing one).

Slightly simplified, this can be stated as

$$\Delta cmd = -\Delta v_s \ ,$$

where v_s is the sensor signal and Δcmd is defined on the domain {0, 1} of cmd. If the v_s is too low (high), i.e. deviates negatively (positively) and, hence, reaches the threshold too early (too late), this causes the command to be set too early (too late), i.e. deviate positively (negatively). The OK model of the inlet valve contains

$$\Delta A = -\Delta cmd \ ,$$

while the outlet valve includes

$$\Delta A = \Delta cmd \ .$$

In summary, based on the OK models of the software and the physical components, the impact of the sensor failure will be determined as for the respective valve failures, in particular overbraking and underbraking.

The relevant failures of the software itself are

- untimely command (which includes "command sent too early", e.g. due to a high threshold value, and "command always"): Δcmd =+ and
- missing command ("command too late or never"): Δcmd = –, triggering the same effects as ΔA =+ (ΔA = –) for the inlet (outlet) valve.

For analogue actuator signals, the deviations generated by the software (either caused by a wrong sensor input or by itself) would be "too high" and "too low".

This may seem to be over-simplified. However, consider that FMEA and also the broader safety analysis is ultimately targeted at determining the failure behavior of the **physical** system and its criticality, and that software bugs are relevant

only with regard to their impact on this, which is totally specified by (deviating) actuator signals. This boils down to faults "untimely/no command" for Boolean signals as discussed above and "signal too high/too low" for analogue ones. Hence, this "physics-centered" perspective makes modeling software faults at this high abstraction level feasible.

7. DISCUSSION

According to the evaluation, so far, the models produced according to the proposed methodology generate the results required by FMEA.

We pointed out that the scope of the models is limited; for instance, they do not capture the impact of air entering the hydraulic circuit. Also, there may be some relevant long-term impact of a fault, which is missed by the system, for instance that a small leakage may not have a dramatic effect immediately, but ultimately causes a relevant drop in the amount of liquid and pressure.

However, the goal of building such tools cannot be to completely replace the human analysis, but rather automatically generate the tables for the vast majority of cases within seconds instead of person days as in the manual process and leave the sophisticated cases to the human experts.

Currently, functional safety analysis gains increased importance, for instance in the automotive industries through the new ISO 26262 standard. This analysis has to go beyond the pure characterization of the physical behavior and also assess its consequences for hazards in various situations, such as collisions, personal damage, and environmental impact. In a different case study, functional safety analysis of a drive train of a truck, described in Struss & Dobi (2013), we extended the analysis in order to derive such conclusions (the risk of collisions with other vehicles, persons, or obstacles in different traffic scenarios) automatically.

ACKNOWLEDGEMENTS

This work benefited from the collaboration with partners in the AUTAS project. Especially, we thank Oskar Dressler for producing a very efficient implementation of the FMEA algorithm.

REFERENCES

Fraracci, A., (2009). *Model-based Failure-modes-and-effects Analysis and its Application to Aircraft Subsystems.* Dissertationen zur Künstlichen Intelligenz DISKI 326, AKA Verlag, ISBN 978-3-89838-326-4, IOS Press, ISBN 978-1-60750-081-0

International Standards Organization (ISO) (2011). "ISO 26262", international Standard ISO/FDIS 26262, 2011. http://www.iso.org/

MIL, (1980). Department of defence USA. *Military standard - procedures for performing a failure mode, effects and criticality analysis*. MIL-STD-1629A, 1980

OCC'M, (2014). OCC'M Software GmbH. *Raz'r Model Editor Version 3*. Interactive Development Environment for Model-based Systems. http://www.occm.de/, (c) 1995-2011

Picardi *et al.*, (2004). C. Picardi, L. Console, F. Berger, J. Breeman, T. Kanakis, J. Moelands, S. Collas, E. Arbaretier, N. De Domenico, E. Girardelli, O. Dressler, P. Struss, B. Zilbermann. *AUTAS: a tool for supporting FMECA generation in aeronautic systems*. In: Proceedings ECAI-2004 Valencia, Spain, pp. 750-754

Pietersma & van Gemund, (2007). J. Pietersma and A.J.C. van Gemund. *Symbolic Factorization of Propagation Delays out of Diagnostic System Models*. In 18th International Workshop on Principles of Diagnosis (DX-07), 2007

Price, C. (2000). *Autosteve: automated electrical design analysis*. In Proceedings ECAI-2000, p.721-725, 2000

Rossi et al., 2008. Rossi, F., van Beek, P., Walsh, T.: *Constraint Programming*. In: van Harmelen, F., Lifschitz, V., and Porter, B. (eds.). Handbook of Knowledge Representation, Elsevier, 2008

SAE, (1993). Society of Automotive Engineers (SAE). *The FMECA process in the Concurrent Engineering (CE) Environment*. SAE AIR4845, 1993

Struss *et al.*, (1997). Struss, P., Sachenbacher, M. Dummert, F.: *Diagnosing a Dynamic System with (almost) no Observations*. Workshop Notes of the 11th International Workshop on Qualitative Reasoning, (QR-97) Cortona, Italy, June 3-6, pp. 193-201, 1997

Struss and Price, (2003). Struss, P., Price, C. *Model-based systems in the automotive industry*. In AI magazine. AAAI Press, Menlo Park (USA) 2003, pp.17-34

Struss, P., (2004). *Models of Behavior Deviations in Model-based Systems*. In. Proceeding of ECAI-2004 Valencia, Spain, pp. 883-887

Struss et al., (2011). Struss, P., Fraracci, A., Nyga, D. : *An Automated Model Abstraction Operator Implemented in the Multiple Modeling Environment MOM*. In: 25th International Workshop on Qualitative Reasoning, Barcelona, Spain, 2011

Struss, P. (2013). *Model-based Analysis of Embedded Systems: Placing it upon its Feet instead of on its Head - An Outsider's View* - In: 8th International Conference on Software Engineering and Applications (ICSOFT-EA 2013), Reykjavik, Iceland, July 29-31 2013

Struss, P. & Dobi, S. (2013). *Automated Functional Safety Analysis of Vehicles Based on Qualitative Behavior Models and Spatial Representations* - In: The 24th International Workshop on Principles of Diagnosis (DX-2013). Jerusalem, Israel/Palestine, Oct. 2013, pp 85-91, http://www.dx-2013.org/proceedings.php

Author Index

A

Abdelwahed, Sherif 207
Acuña, David E. 338
Adams, Douglas 561
Agarwala, Vinod S. 298
Ahn, Hyung-il 664
Al Tamimi, Abdallah 278
Allen, David 331
Ardakani, Hossein D. 167
Ardis, Paul A. 252
Avram, Remus C 217

B

Bahi, J.M. 681
Bailey, Kevin D. 402
Balaban, Edward 142
Battat, Mor 161
Bechhoefer, Eric 118
Bellas, Anastasios 415
Bellazzi, Alberto 423
Bertram, Torsten 309
Bird, Ross 29
Blair, Mark 62
Bliznyuk, Alexander 151
Bole, Brian 502
Bortman, Jacob 151, 161
Bower, Gregory 29
Brat, Guillaume 176
Bregon, Anibal 128, 347
Brown, B. 70
Brown, Douglas W. 287, 298, 748
Bryant, Michael D. 109, 741
Bussey, Howard E. 252

C

Caminhas, Walmir 392
Campbell, Jacob 217
Carter-Journet, Katrina 519
Catterson, Victoria M. 70, 713
Celaya, José R. 142
Chanthery, Elodie 51
Chen, Cen 705
Chinnam, Ratna Babu 604
Choi, Joo-Ho 494
Châtelet, Eric 41
Connolly, Richard J. 298, 748
Cox, Justin 102
Crabb, Charles 451

D

Dadon, Ido 151
Daigle, Matthew 76, 128, 142, 347, 502
Daoud, Sami 402
de Kleer, Johan 589
Donmez, M. Alkan 576
Dubey, Abhishek 207, 238
Dutta, Partha 2

E

Echauz, Javier 287
Edge, Joseph C. 367
Elghazel, W. 681

F

Falgout, Troy A. 519
Ferrio, Kyle B. 754
Filev, Dimitar P. 604
Forrester, David 378
Fraracci, Alessandro 775
Frisk, Erik 92
Frith, Peter 378

G

Gabbar, Hossam A. 41
Galloway, Grant S. 70, 713
Gao, Robert X. 658
Garvan, Margaret R. 287, 298
Gaudel, Quentin 51
Gellot, F. 597
Giannakopoulou, Dimitra 176
Giljohann, Sebastian 692
Godwin, Jamie L. 20
Goebel, Kai 76, 356, 533
Goh, Chi Keong 2
Gollnick, Volker 435
Gomez-Gonzalez, Vanesa 188
Goodman, Douglas L. 754
Gorospe, George 76, 142
Guyeux, C. 681
Guépié, Blaise 762

H

Ha, Jong Moon 494
Haglind, Fredrik 511
Hakem, M. 681
Hardman, William 102
Harrison, E. 70
He, David 118, 732
Hecke, Brandon Van 118, 732

Heuermann-Kuehn, Ludger 519
Hing, James 62
Hochstein, Axel 664
Holbøll, Joachim 725
Hopkins, Brad 648
Hoyle, Chris 176
Hölzel, Nico B. 435

I

Iamsumang, Chonlagarn 320
Imanian, Anahita 10
Inacio, Maurilio 392

J

Jacazio, Giovanni 423
Jensen, Bogi Bech 725
Jin, Chao 167
Johnson, Preston 465

K

Kale, Amit A. 519
Kandepu, Rambabu 511
Karsai, Gabor 188, 238
Kessler, Seth 102
Kim, Nam Ho 494
Kinloch, Gordon 402
Klein, Renata 151, 161
Klingauf, Uwe 692
Kodali, Anuradha 198
Kogan, Gideon 161
Kosson, David 561
Krysander, Mattias 92
Kulkarni, Chetan S. 76, 502
Kurtoglu, Tolga 589
Kwuimy, C. A. Kitio 484
Kählert, Alexander 692

L

Lacaille, Jérôme 415
Lam, Jack 623
Lambert, Nicholas A. 754
Larsson, Emil 92
Laskowski, Bernard 298, 748
Lecoeuche, Stéphane 762
Lee, Jay 167
Leem, Sang Hyuck 494
Lemos, Andre 392
Leung, Ying Tat 664
Li, Honglei 287, 298, 705
Li, Jiaming 287, 546
Liang, Xihui 568
Liao, Linxia 589
Lim, Pin 2
Liu, Chen-Ching 238
Liu, Zhiliang 86
Liu, Zongchang 167
Louajri, Hanane 265
Love, Craig 713
Lowry, Michael 188

M

Mahadevan, Nagabhushan 188, 238
Mahadevan, Sankaran 207, 561
Maino, Bruno 423
Makki, Imad H. 604
Malere, João Pedro P. 676
Maraini, Dan 648
Marangé, P. 597
Marhadi, Kun S. 225, 725
Martin, Everard 62
Matthews, Peter 20
McGhee, M. J. 70
Medjaher, K. 681
Mehrpouyan, Hoda 176
Mendelovich, Matan 161
Mihaylov, Gueorgui 423
Mijatovic, Nenad 725
Mills, Andrew R. 367
Minhas, Raj 589
Modarres, Mohammad 10, 278, 320
Mokhtar, Maizura 367
Morton, Brogan 635
Mosleh, Ali 320

N

Nannapaneni, Saideep 207
Narasimhan, Sriram 142
Nataraj, C. 484
Neema, Sandeep 207
Nenadic, Nenad G. 252
Nostrand, Thomas J. 118, 635

O

Oliva, Javier A. 309
Ompusunggu, Agusmian P. 167
Orchard, Marcos E. 338
Ozevin, Didem 102

P

Park, Jungho 494
Pellerey, Franco 423
Petreczky, Mihaly 762
Petré, Fredrik 167
Philippot, A. 597
Pierobon, Leonardo 511
Pérez, Aramis 338
Pétin, J.F. 597

R

Ramasso, Emmanuel 612
Rangarajan, Arvind 589
Ribot, Pauline 51
Riera, B. 597
Robb, Andrew 713
Robinson, Peter 188, 198
Roychoudhury, Indranil 128, 142, 347

S

Salhi, Leila 553
Samadani, M. 484
Sanders, Yitschak 161
Sankararaman, Shankar 356, 533, 623
Saxena, Abhinav 533, 612
Sayed-Mouchaweh, Moamar 265
Schilling, Thomas 435
Schlanbusch, Rune 511
Schumann, Johann 188
Seidel, Andrew 648
Shahidi, Parham 648
Shevach, Glenn 62
Silva, Jorge F. 338
Skrimpas, Georgios Alexandros 225, 725
Sorli, Massimo 423
Soulier, Eddie 41
Spiegel, Cody 635
Srivastava, Anurag 238
Stecki, Chris 553
Stecki, Jacek 553
Stewart, Bryan 623
Struss, Peter 775
Sweeney, Christian Walsted 725
Sweet, Adam 142

T

Tan, Kay Chen 2
Teubert, Christopher 356
Thurston, Michael G. 252
Timmons, Alan 102
Traore, Moussa 41
Tseng, Fling 604
Tumer, Irem Y. 176

V

Vachtsevanos, George J. 287, 298, 546, 705
Venetsky, Larry 62
Vianna, Wlamir O. L. 676
Villeburn, Michal J. 402
Vogl, Gregory W. 576

W

Wang, Peng 658
Wang, Wenyi 378
Wang, Xiaofeng 473
Wang, Xiaoyun 770
Waters, Nicholas 748
Weiss, Brian A. 576
Wheelock, John 62

Y

Ye, Xuerong 705
Yoon, Jae 118, 732
Youn, Byeng D. 494

Z

Zerhouni, N. 681
Zhang, Bin 473
Zhang, Longlong 86
Zhang, Xiaodong 217
Zhao, Tingdi 770
Zhu, Junda 118, 635
Zook, Jonathan 29
Zuo, Ming J. 86, 568
Zurcher, Derick 519