UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1996

SUMMER RESEARCH EXTENSION PROGRAM FINAL REPORTS

VOLUME 5

ARNOLD ENGINEERING CENTER
AIR LOGISTICS CENTERS

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major Linda Steel-Goodwin

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Johnetta Thompson

Program Administrator
Rebecca Kelly-Clemmons

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1996

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0707

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. R
	December, 1996	

4. TITLE AND SUBTITLE
1996 Summer Research Program (SRP), Summer Research Extension Program (SREP), Final Report, Volume 5, Arnold Eng. Development Center (AEDC) and Air Logistics Centers

5. FUNDING NUMBERS
F49620-93-C-0063

6. AUTHOR(S)
Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research & Development Laboratories (RDL)
5800 Uplander Way
Culver City, CA 90230-6608

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research (AFOSR)
801 N. Randolph St.
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The United States Air Force Summer Research Program (SRP) is designed to introduce university, college, and technical institute faculty members to Air Force research. This is accomplished by the faculty members, graduate students, and high school students being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory (AFRL) Technical Directorates and Air Force Air Logistics Centers (ALC). AFOSR also offers its research associates (faculty only) an opportunity, under the Summer Research Extension Program (SREP), to continue their AFOSR-sponsored research at their home institutions through the award of research grants. This volume consists of a listing of the participants for the SREP and the technical report from each participant working at the Arnold Engineering Development Center (AEDC) and Air Logistics Centers.

14. SUBJECT TERMS
Air Force Research, Air Force, Engineering, Laboratories, Reports, Summer, Universities, Faculty, Graduate Student, High School Student

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	**PR** - Project
G - Grant	**TA** - Task
PE - Program Element	**WU** - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with....; Trans. of....; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD -	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE -	See authorities.
NASA -	See Handbook NHB 2200.2.
NTIS -	Leave blank.

Block 12b. Distribution Code.

DOD -	Leave blank.
DOE -	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA -	Leave blank.
NTIS -	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self- explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

PREFACE

This volume is part of a five-volume set that summarizes the research of participants in the 1996 AFOSR Summer Research Extension Program (SREP.) The current volume, Volume 1 of 5, presents the final reports of SREP participants at Armstrong Laboratory. Volume 1 also includes the Management Report.

Reports presented in this volume are arranged alphabetically by author and are numbered consecutively – e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3, with each series of reports preceded by a 35 page management summary. Reports in the five-volume set are organized as follows:

VOLUME	TITLE
1	Armstrong Laboratory
2	Phillips Laboratory
3	Rome Laboratory
4A	Wright Laboratory
4B	Wright Laboratory
5	Arnold Engineering Development Center Air Logistics Centers

1996 SREP FINAL REPORTS

Armstrong Laboratory

VOLUME 1

Report #	Report Title Author's University	Report Author
1	Chlorinated Ethene Transformation, Sorption & Product Distr in Metallic Iron/Water Systems: Effect of Iron Properties Washington State University, Pullman, WA	Dr. Richelle M Allen-King Dept. of Geology AL/EQ
2	Dynamically Adaptive Interfaces: A Preliminary Investigation Wright State University, Dayton, OH	Dr. Kevin B Bennett Dept. of Psychology AL/CF
3	Geographically Distributed Collaborative Work Environment California State University, Hayward, CA	Dr. Alexander B Bordetsky Dept. Decesion Sciences AL/HR
4	Development of Fluorescence Post Labeling Assay for DNA Adducts: Chloroacetaldeh New York Univ Dental/Medical School, New York, NY	Dr. Joseph B Guttenplan Dept. of Chemistry AL/OE
5	The Checkmark Pattern & Regression to the Mean in Dioxin Half Life Studies University of South Alabama, Mobile, AL	Dr. Pandurang M Kulkarni Dept. of Statistics AL/AO
6	Determination of the Enzymatic Constraints Limiting the Growth of Pseudomonas University of Dayton, Dayton, OH	Dr. Michael P Labare Dept. of Marine Sciences AL/HR
7	Tuned Selectivity Solid Phase Microextraction Clarkson University, Potsdam, NY	Dr. Barry K Lavine Dept. of Chemistry AL/EQ
8	A Cognitive Engineering Approach to Distributed Team Decision Making During University of Georgia, Athens, GA	Dr. Robert P Mahan Dept. of Psychology AL/CF
9	Repetative Sequence Based PCR: An Epidemiological Study of a Streptococcus Stonehill College, North Easton, MA	Dr. Sandra McAlister Dept. of Biology AL/CF
10	An Investigation into the Efficacy of Headphone Listening for Localization of Middle Tennessee State University, Murfreesbord, TN	Dr. Alan D. Musicant Dept. of Psychology AL/CF
11	The Neck Models to Predict Human Tolerance in a G-Y CUNY-City College, New York, NY	Dr. Ali M. Sadegh Dept. of Mech Engineering AL/CF

1996 SREP FINAL REPORTS

Armstrong Laboratory

VOLUME 1 (cont.)

Report #	Report Title Author's University	Report Author
12	Tracer Methodology Development for Enhanced Passive Ventilation for Soil University of Florida, Gainesville, FL	Dr. William R. Wise Dept. of Civil Engineering AL/EQ
13	Application of a Distribution-Based Assessment of Mission Readiness System for the Evaluation of Personnel Training Texas A&M University, College Station, TX	Dr. David J. Woehr Dept. of Psychology AL/HR
14	Electrophysiological, Behaviorial, and Subjective Indexes of Workload when Performing Multiple Tasks Washington State University, Pullman, WA	Ms. Lisa Fournier Dept. of Psychology AL/CF
15	Methods for Establishing Design Limits to Ensure Accomodation for Ergonomic Design Miami University, Oxford, OH	Ms. Kristie Nemeth Dept. of Psychology AL/HR

1996 SREP FINAL REPORTS

Phillips Laboratory

VOLUME 2

Report #	Report Title Author's University	Report Author
1	Experimental Study of the Tilt Angular Anisopalantaic Correlation & the Effect Georgia Tech Research Institute, Atlanta, GA	Dr. Mikhail Belen'kii Dept. of Electro Optics PL/LI
2	Performance Evaluations & Computer Simulations of Synchronous & Asynchronous California State University, Fresno, CA	Dr. Daniel C. Bukofzer Dept. of Elec Engineering PL/VT
3	MM4 Model Experiments on the Effects of Cloud Shading Texas Tech University, Lubbock, TX	Dr. Chia-Bo Chang Dept. of Geosciences PL/GP
4	Miniature Laser Gyro consisting in a Pair of Unidirectional Ring Lasers University of New Mexico, Albuquerque, NM	Dr. Jean-Claude M. Diels Dept. of Physics PL/LI
5	Simulations & Theoretical Studies of Ultrafast Silicon Avalanche Old Dominion University, Norfolk, VA	Dr. Ravindra P. Joshi Dept. of Elec Engineering PL/WS
6	Theory of Wave Propagation in a Time-Varying Magnetoplasma Medium & Applications to Geophysical Phenomena University of Massachusetts Lowell, Lowell, MA	Dr. Dikshitulu K. Kalluri Dept. of Elec Engineering PL/GP
7	Thermal Analysis for the Applications of High Power Lasers in Large-Area Materials Processing University of Central Florida, Orlando, FL	Dr. Arvinda Kar Dept. of Engineering PL/LI
8	Analytical Noise Modeling and Optimization of a Phasor-Based Phase Texas Tech University, Lubbock, TX	Dr. Thomas F. Krile Dept. of Elec Engineering PL/LI
9	Mathematical Modeling of Thermionic-AMTEC Cascade System for Space Power Texas Tech University, Lubbock, TX	Dr. M. Arfin K. Lodhi Dept. of Physics PL/VT
10	Preparation & characterization of Polymer Blends Ohio State University, Columbus, OH	Dr. Charles J. Noel Dept. of Chemistry PL/RK
11	Evaluation of Particle & Energy Transport to Anode, Cathode University of Texas-Denton, Denton, TX	Dr. Carlos A. Ordonez Dept. of Physics PL/WS
12	Analysis of the Structure & Motion of Equatorial Emission Depletion Bands Using Optical All-Sky Images University of Massachusetts Lowell, Lowell, MA	Dr. Ronald M. Pickett Dept. of Psychology PL/GP

1996 SREP FINAL REPORTS

Phillips Laboratory

VOLUME 2 (cont.)

Report #	Author's University	Report Author
13.	On the Fluid Dynamics of High Pressure Atomization in Rocket Propulsion University of Illinois-Chicago, Chicago, IL	Dr. Dimos Poulikakos Dept. of Mech Engineering PL/RK
14	Gigahertz Modulation & Ultrafast Gain Build-up in Iodine Lasers University of New Mexico, Albuquerque, NM	Dr. W. Rudolph Dept. of Physics PL/LI
15	Inversion of Hyperspectral Atmospheric Radiance Images for the Measurement of Temperature, Turbulence, and Velocity University of New Mexico, Albuquerque, NM	Dr. David Watt Dept. of Mech Engineering PL/GP

1996 SREP FINAL REPORTS

Rome Laboratory

VOLUME 3

Report #	Author's University	Report Author
1	Performance Analysis of an ATM-Satellite System Florida Atlantic University, Boca Raton, FL	Dr. Valentine Aalo Dept. of Elec Engineering RL/C3
2	Reformulatiang Domain Theories to Improve their Computational Usefulness Oklahoma State University, Stillwater, OK	Dr. David P. Benjamin Dept. of Comp Engineering RL/C3
3	An Analysis of the Adaptive Displaced Phase Centered Antenna Lehigh University, Bethlehem, PA	Dr. Rick S. Blum Dept. Elec Engineering RL/OC
4	Effect of Concatenated Codes on the Transport of ATM-Based Traffic California Polytechnic State, San Luis Obispo, CA	Dr. Mostafa Chinichian Dept. of Engineering RL/C3
5	Development of Efficient Algorithms & Software Codes for Lossless and Near-Lossless Compression of Digitized Images Oakland University, Rochester, MI	Dr. Manohar K. Das Dept. Elec Engineering RL/IR
6	Mode-Locked Fiber Lasers Rensselaer Polytechnic Institution, Troy, NY	Dr. Joseph W. Haus Dept. of Physics RL/OC
7	Magnitude & Phase Measurements of Electromagnetic Fields Using Infrared University of Colorado, Colorado Springs, CO	Dr. John D. Norgard Dept. Elec Engineering RL/ER
8	Image Multiresolution Decomposition & Progressive Transmision Using Wavelets New Jersey Institute of Technology, Newark, NJ	Dr. Frank Y. Shih Dept. of Comp Science RL/IR
9	Investigation of Si-Based Quantum Well Intersubband Lasers University of Massachusetts-Boston, Boston, MA	Dr. Gang Sun Dept. of Physics RL/ER
10	Numerical Study of Bistatic Scattering from Land Surfaces at Grazing Incidence Oklahoma State University, Stillwater, OK	Dr. James C. West Dept. of Elec Engineering RL/ER

1996 SREP FINAL REPORTS

Wright Laboratory

VOLUME 4A

Report #	Author's University	Report Author
1	Barrel-Launched Adaptive Munition Experimental Round Research Auburn University, Auburn, AL	Dr. Ronald M. Barrett Dept. of Aerospace Eng WL/MN
2	Modeling & Design of New Cold Cathode Emitters & Photocathodes University of Cincinnati, Cincinnati, OH	Dr. Marc M. Cahay Dept. of Elec Engineering WL/EL
3	Unsteady Aerodynamics University of California-Berkeley, Berkeley, CA	Dr. Gary Chapman Dept. of Aerospace Eng WL/MN
4	Characteristics of the Texture Formed During the Annealing of Copper Plate University of Nebraska-Lincoln, Lincoln, NE	Dr. Robert J. DeAngelis Dept. of Mech Engineering WL/MN
5	Development of Perturbed Photoreflectance, Implementation of Nonlinear Optical Parametric Devices Bowling Green State University	Dr. Yujie J. Ding Dept. of Physics WL/EL
6	Computations of Drag Reduction & Boundary Layer Structure on a Turbine Blade with an Oscillating Bleed Flow University of Dayton, Dayton, OH	Dr. Elizabeth A. Ervin Dept. of Mech Engineering WL/PO
7	Low Signal to Noise Signal Processor for Laser Doppler Velocimetry North Carolina State University, Raleigh, NC	Dr. Richarad D. Gould Dept. of Mech Engineering WL/PO
8	Modeling & Control for Rotating Stall in Aeroengines Louisiana State University, Baton Rouge, LA	Dr. Guoxiang Gu Dept. of Elec Engineering WL/FI
9	Scaleable Parallel Processing for Real-time Rule-Based Decision Aids University of Missouri-Columbia, Columbia, MO	Dr. Chun-Shin Lin Dept. of Elec Engineering WL/FI
10	Quantitative Image Location & Processing in Ballistic Holograms University of West Florida, Pensacola, FL	Dr. James S. Marsh Dept. of Physics WL/MN
11	Experimental & Computational Investigation of Flame Suppression University of North Texas, Denton, TX	Dr. Paul Marshall Dept. of Chemistry WL/ML
12	Investigations of Shear Localization in Energetic Materials Systems University of Notre Dame, Notre Dame, IN	Dr. James J. Mason Dept. of Aerospace Eng WL/MN

1996 SREP FINAL REPORTS

Wright Laboratory

VOLUME 4A (cont.)

Report #	Author's University	Report Author
13	A Time Slotted Approach to Real-Time Message Scheduling on SCI University of Nebraska-Lincoln, Lincoln, NE	Dr. Sarit Mukherjee Dept. of Comp Engineering WL/AA
14	Dielectric Resonator Measurements on High Temperature Superconductor (HTS) Wright State University, Dayton, OH	Dr. Krishna Naishadham Dept. Elec Engineering WL/ML
15	Modeling of Initiation & Propagation of Detonation Energetic Solids University of Notre Dame, Notre Dame, IN	Dr. Joseph M. Powers Dept. of Aerospace WL/MN
16	Robust control Design for Nonlinear Uncertain Systems by Merging University of Central Florida, Orlando, FL	Dr. Zhihua Qu Dept. of Elec Engineering WL/MN

1996 SREP FINAL REPORTS

Wright Laboratory

VOLUME 4B

Report #	Author's University	Report Author
17	HELPR: A Hybrid Evolutionary Learning System Wright State University, Dayton, OH	Dr. Mateen M. Rizki Dept. of Comp Engineering WL/AA
18	Virtual Materials Processing: automated Fixture Design for Materials Southern Illinois University-Carbondale, IL	Dr. Yiming K. Rong Dept. of Technology WL/ML
19	A Flexible Architecture for Communication Systems (FACS): Software AM Radio Wright State University, Dayton, OH	Dr. John L. Schmalzel Dept. of Engineering WL/AA
20	A Design Strategy for Preventing High Cycle Fatigue by Minimizing Sensitivity of Bladed Disks to Mistuning Wright State University, Dayton, OH	Dr. Joseph C. Slater Dept. of Mech Engineering WL/FI
21	Growth of Silicon Carbide Thin Films by Molecular Beam Epitaxy University of Cincinnati, Cincinnati, OH	Dr. Andrew J. Steckl Dept. of Elec Engineering WL/FI
22	Performance of Iterative & Noniterative Schemes for Image Restoration University of Arizona, Tucson, AZ	Dr. Malur K. Sundareshan Dept. of Elec Engineering WL/MN
23	Improving the Tribological Properties of Hard TiC Coatings University of New Orleans, New Orleans, IA	Dr. Jinke Tang Dept. of Physics WL/ML
24	Development of Massively Parallel Epic Hydrocode in Cray T3D Using PVM Florida Atlantic University, Boca Raton, FL	Dr. Chi-Tay Tsai Dept. of Mech Engineering WL/MN
25	Supramolecular Multilayer Assemblies w/Periodicities in a Submicron Range Western Michigan University, Kalamazoo, MI	Dr. Vladimir V. Tsukruk Dept. of Physics WL/ML
26	Distributed Control of Nonlinear Flexible Beams & Plates w/Mechanical & Temperature Excitations University of Kentucky, Lexington, KY	Dr. Horn-Sen Tzou Dept. of Mech Engineering WL/FI
27	A Progressive Refinement Approach to Planning & Scheduling University of Colorado-Denver, Denver, CO	Dr. William J. Wolfe Dept. of Comp Engineering WL/MT
28	Development of a New Numerical Boundary condition for Perfect Conductors University of Idaho, Moscow, OH	Dr. Jeffrey L. Young Dept. of Elec Engineering WL/FI

1996 SREP FINAL REPORTS

Wright Laboratory

VOLUME 4B (cont.)

Report #	Author's University	Report Author
29	Eigenstructure Assignment in Missile Autopilot Design Using a Unified Spectral Louisiana State University, Baton Rouge, LA	Dr. Jianchao Zhu Dept. of Elec Engineering WL/FI
30	Design & Implementation of a GNSS Software Radio Receiver Ohio University, Athens, OH	Dr. Dennis M. Akos Dept. of Elec Engineering
31	Experimental & Numerical Study of Localized Shear as an Initiation Mechanism University of Notre Dame, Notre Dame, IN	Mr. Richard J. Caspar Dept. of Aero Engineering WL/MN
32	A Molecular-Level view of Solvation in Supercritical Fluid Systems State University of New York – Buffalo, Buffalo, NY	Ms. Emily D. Niemeyer Dept. of Chemistry WL/PO
33	Initiation of Explosives by High Shear Strain Rate Impact University of Notre Dame, Notre Dame, IN	Mr. Keith M. Roessig Dept. of Aero Engineering WL/MN

1996 SREP FINAL REPORTS

VOLUME 5

Report #	Author's University	Report Author
	Arnold Engineering Development Center	
1	**Facility Health Monitoring & Diagnosis** **Vanderbilt University, Nashville, TN**	**Dr. Theodore Bapty** **Dept. of Elec Engineering** **AEDC**
	Air Logistic Centers	
2	**Fatigue Crack Growth Rates in Naturally-Coroded Aircraft Aluminum** **University of Oklahome, Norman, OK**	**Dr. James D. Baldwin** **Dept. of Mech Engineering** **OCALC**
3	**A Novel Artificial Neural Network Classifier for Multi-Modal** **University of Toledo, Toledo, OH**	**Dr. Gursel Serpen** **Dept. of Elec Engineering** **OOALC**
4	**Development of a Cost-Effective Organizational Information System** **West Virginia University, Morgantown, WV**	**Dr. Michael D. Wolfe** **Dept. Mgmt Science** **SAALC**
5	**Implementation of a Scheduling Software w/Shop Floor Parts Tracking Sys** **University of Wisconsin-Stout, Menomonie, WI**	**Dr. Norman D. Zhou** **Dept. of Technology** **SMALC**
6	**Development of a High Performance Electric Vehicle Actuator System** **Clarkson University, Potsdam, NY**	**Dr. James J. Carroll** **Dept. Elec Engineering** **WRALC**

FACILITY MONITORING AND DIAGNOSTICS

Dr. Theodore A. Bapty
Research Assistant Professor
Measurement and Computing Systems Laboratory
Department of Electrical and Computer Engineering

Vanderbilt University
Box 1649, Station B
Nashville, TN 37235-0000

Final Report for:
Summer Faculty Research Program
Arnold Engineering Development Center,
Arnold AFB, TN

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Arnold Engineering Development Center

December 1996

FACILITY MONITORING AND DIAGNOSTICS

Theodore A. Bapty
Research Assistant Professor
Department of Electrical and Computer Engineering
Vanderbilt University

Abstract

Arnold Engineering Development Center operates and maintains the country's largest collection of aerospace ground test facilities. Ensuring the proper functionality of this equipment is a difficult task. This project developed tools to assist in the diagnostics and monitoring of the rotating machinery that is critical to AEDC operations. The major components of the work were the interfacing of a telemetry unit to a real-time instrumentation parallel processor and the implementation of specialized algorithms for vibration monitoring.

FACILITY MONITORING AND DIAGNOSTICS

Theodore A. Bapty

Introduction

Arnold Engineering Development Center is charged with supporting high fidelity, high reliability ground testing of aerospace systems. Full-scale systems can be tested at simulated flight conditions, replicating altitudes, temperatures, pressures, air mass-flows, etc. that are encountered by the test article under standard operation. The primary advantages of ground testing are safety, control, and access to measurements that would not be available in flight testing.

In order to produce flight conditions at the interfaces to the test article require a complex array of equipment.. For example, simulating conditions for engine testing requires a city-block-long facility employing many very large electric motors, compressors, heaters and coolers, and a network of valves and diffusers. This equipment is extremely expensive to purchase, and must be available constantly to support the needs of defense system development.

Maintaining these facilities is a challenge, given the need for reliability and the sheer magnitude of the facilities. The objective of this research is the development of tools to assist in the monitoring and maintenance of these facilities.

The specific types of problems we are interested in detecting in this research are primarily vibrational. The vibrations of the machinery components can be used to detect bearing problems, shaft imbalances, and gear problems.

Research Objectives

In order to continuously assess the health of the rotating machinery, we must perform two primary functions:

- **Acquisition of Data**. The equipment must be instrumented with sensors to measure the relevant parameters. Sensors are routinely available to measure strain and accelerations (strain gauges and accelerometers). The operational problem in acquiring the data becomes the connection of the sensors to the data acquisition & processing system. Many of the measurement positions are on rotating components. Typically, slip rings are used to connect the sensors to the outside world, however this

can introduce expense in installing the sensors and will superimpose noise on the data. Often, the diagnostics system is not to be a permanent fixture, so expensive installations are not viable. For this project, we have chosen to take advantage of results from another AEDC project that implemented a telemetry system. This system was intended for telemetry of data from short events, high-velocity G-Range experiments. For the diagnostics application, we require a continuous stream of data.

- **Analysis of Data.** The measured signals contain information that is not readily apparent by looking at time-domain, oscilloscope-style diagrams. The system must offer analysis algorithms that can extract features from the data to assist in the diagnostics. The primary analysis tools for analysis reside in the frequency domain. This information can be used in several ways:
    - The presence of specific frequencies can be used to determine aberrant vibrations.
    - The trending of frequencies and amplitudes over time can indicate a development or worsening of a problem.
    - Tracking these frequencies versus an independent parameter can be used to enhance small spectral effects and can indicate causal effects within the system.
    - Phase-domain processing can be used to confirm cause-and-effect relationships between observed frequencies and multiples of those frequencies.

Research Results

The size of the project precluded the development of a full telemetry system. An in-house research project at AEDC had developed a telemetry module for instrumenting projectiles in the high-speed G-Range at AEDC. The requirements of the telemetry system developed for this application were significantly different from the machinery monitoring system. The G-Range system needed to operate for several milliseconds, recording a burst of data for storage. On-line analysis in the G-Range application was not necessary, due to the operational time.

The diagnostics effort requires a continuous data stream from multiple sensor channels. Very large storage buffers are needed to accumulate meaningful data. More importantly, the data must be continuously processed to watch for developing phenomena.

The G-Range telemetry system consists of several distinct components:

1. The battery operated digitizing subsystem. This consists of a set of analog amplifiers, a signal multiplexing system, an analog-to-digital converter, and a controller circuit. Special sensor interface circuits are added as needed. The output of this subsystem is a digital bit stream with a unique synchronization marker and the channel number encoded. The data format supports from 1 to 64 analog channels.

2. The battery operated data transmitter subsystem. This component accepts the digital signal from the digitizing subsystem and broadcasts out via an antenna as an Radio-Frequency(RF) signal. The range of the system is limited to limit interference with other devices.
3. The data receiver subsystem. This component receives the RF signal and reproduces the digital bit stream from the digitizing subsystem.
4. The data logging/computer interface subsystem. This component receives the bit stream, detects the synchronization sequence, and aligns the data. The G-Range interface contains DRAM memory for buffering the data as it is received. After acquisition, the subsystem interfaces with a standard parallel EPP port for data retrieval to a portable computer. Logging of the data is supported for short times only, limited by the size of the DRAM memory. Continuous acquisition is not supported.

Of these components, the first 3 were usable for this project directly without modification. Combining these components, we get a digital bit stream with the format as shown in figure 1.

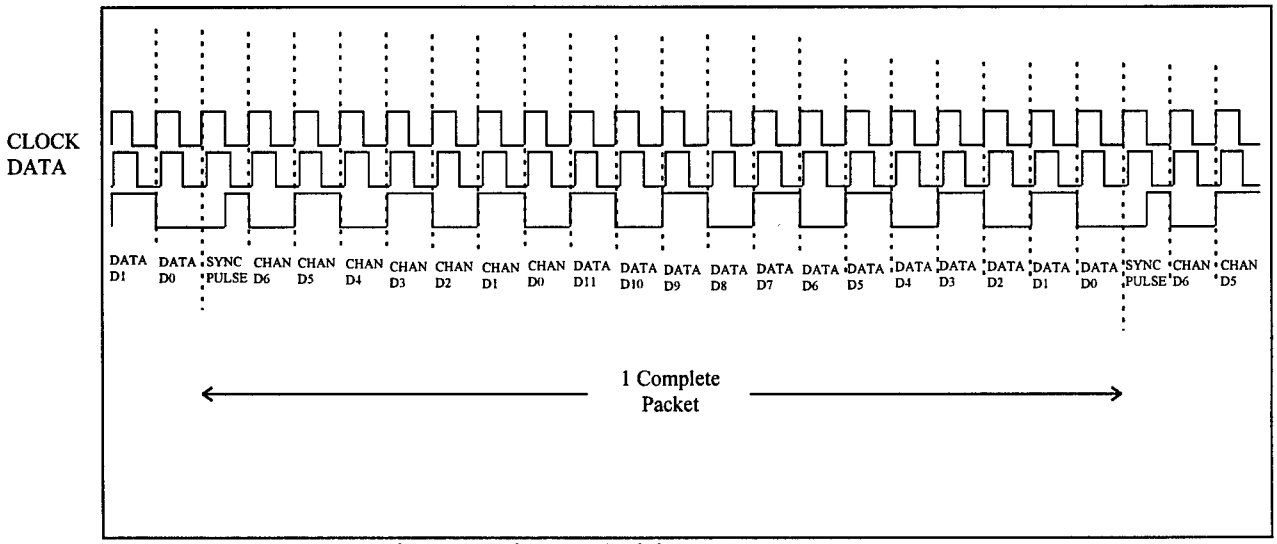


Figure 1: Telemetry Serial Data Format

Since we are interested in continuous processing, with algorithms that require significant computation, we must have an interface to a high performance signal processing system. The DSP system of choice is the TMS320C40. This processor was chosen because it was already in use in several high performance signal analysis systems at AEDC, such as the Computer Assisted Dynamic Data Analysis System(CADDMAS). The C4x processors have 4-6 high-speed communication ports, that accept byte-wide data at rates up to 20 megabytes per second.

Telemetry Interface Circuit Description

To translate between the telemetry serial format and the C40 communication port, a circuit is required. In addition to translating formats, the device must buffer data to allow for software latencies without losing data.

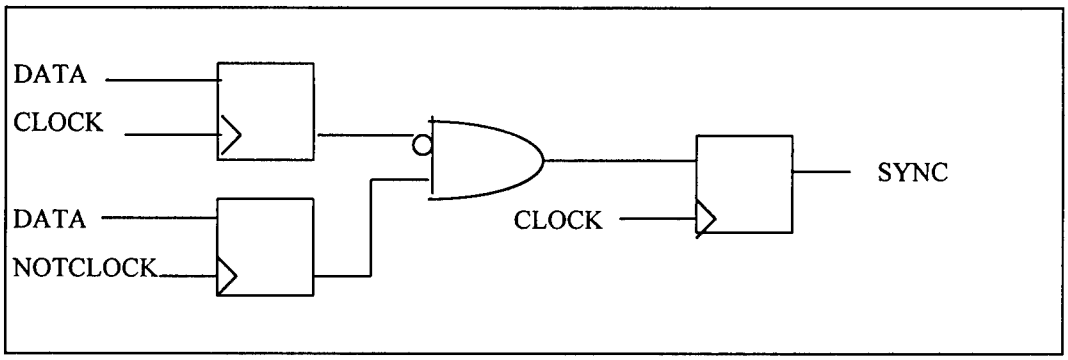
Synchronization

The first step in the design is to detect the synchronization sequence. Two signals are available from the telemetry receiver, CLOCK and DATA. CLOCK is a 2 MHz square wave, marking the edges of the data bits on each rising edge. DATA contains the telemetry information, changing levels at the rising CLOCK edge. The synch bit is different from all standard bits in that it is a half-width pulse, starting low for the first (high) half of the clock and rising for the second half of the clock with the fall of the CLOCK signal. A synchronization pulse can be generated by the following procedure:

1. Latch the DATA on the rising edge of CLOCK (CLOCKLATCH)
2. Latch the DATA on the falling edge of CLOCK (NOTCLOCKLATCH)
3. Generate the function (!CLOCKLATCH & NOTCLOCKLATCH)
4. Latch this function on the rising edge of CLOCK.

This circuit, as shown in figure 2, reliably detects the synchronization without requiring generation of higher clock frequencies, a procedure which can be noise sensitive.

Figure 2: Generation of SYNC



Data Reception

With a valid sync signal, the serial data bit stream can be captured and aligned. The input data format is a sequential stream of bits, beginning with the most significant bit of the channel number, and proceeding down to the LSB of the 12 bit data word. The output requires a sequence of bytes, beginning with the least significant byte.

The translation is carried out using a set of cascaded shift registers, using CLOCK as the clock and DATA as the input. When a SYNC is detected, the shift registers are transferred into a register on the following

clock. During the transfer, the word is shifted down by 2 bit positions to account for the SYNC bit and the delay of one CLOCK for sync detection.

After transfer to the register, 4 bytes are loaded into the external FIFO (if space is available), least significant byte first. Since the C40 operates with 32 bit words, the telemetry data is extended to 32 bits, with a marker byte of 0xD4 to ensure synchronization. A state machine within the FPGA controls this transfer.

Data Transfer

An independent state machine controls the transmission of the data to the C40. The C40 uses an asynchronous protocol implemented with the STRB and ACK lines. The protocol is as follows:

1. The transmitting device asserts the first 8 bits of data and lowers STRB.
2. Detecting STRB, the receiving device acquires the data and asserts ACK.
3. Detecting ACK, the transmitting device de-asserts STRB.
4. The receiving device de-asserts ACK

This sequence repeats 3 more times to transmit 4 bytes.

The telemetry interface implements this protocol, using the FIFO as the data source. This protocol is implemented by a state machine within the FPGA. The initiation of a byte transfer is triggered by a FIFO_NOT_EMPTY flag.

The cable to the C40 can operate up to 20 MHz, and must be treated as a transmission line. To match the impedance of the cable, series resistors are inserted in each data and control line. Data transfer has been tested and proven reliable with 3 foot long unshielded cables.

Telemetry Interface Circuit Description

The circuits described are implemented in a Xilinx Field Programmable Gate Array(FPGA). The FPGA has the advantage that the internal circuitry can be redefined an unlimited number of times. This allows a great deal of flexibility in implementing diverse functions. The FPGA used, the Xilinx 4003e, provides approximately 3000 gate equivalents, allowing a complex system to be implemented on-chip. The programming of this chip is done on power-up, by the host PC.

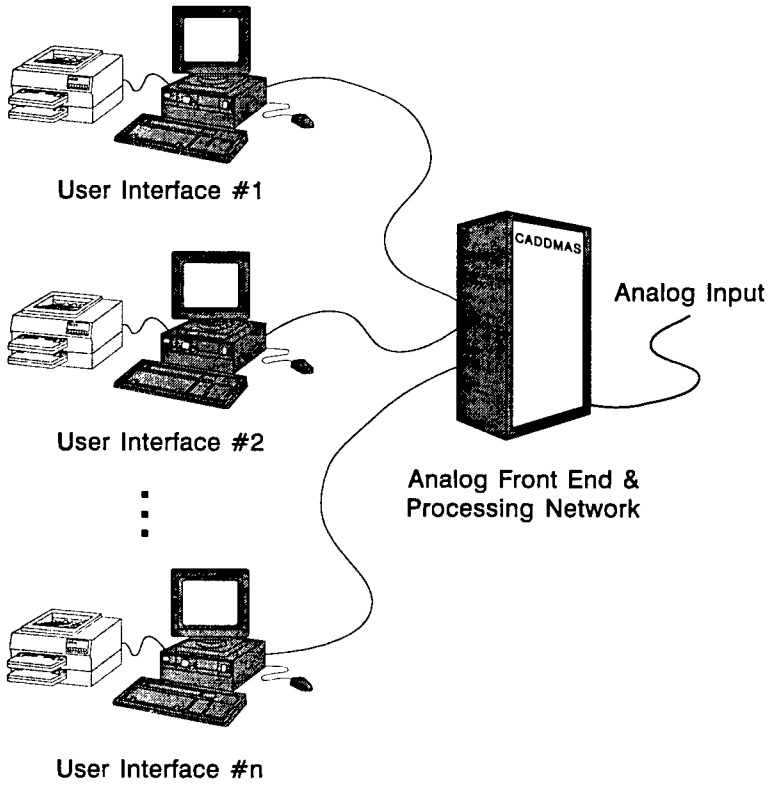
The telemetry interface board uses the same physical printed circuit board as the CADDMAS analog-to-digital board developed. This board typically has 8 analog-to-digital chips installed. Since these chips have serial data outputs, the data paths required for connection to the telemetry receiver are already built into the board.

The output to C40 is already built into the board, with FIFO and C40 compatible connector. These sections were not modified for this application.

CADDMAS Hardware Overview

The telemetry unit produces data continuously at 100K samples per second. Significant processing power is required to analyze the data continuously, in real-time. This project takes advantage of the infrastructure developed for the Computer Assisted Dynamic Data Monitoring and Analysis System (CADDMAS) to provide a platform for this processing.

The CADDMAS system consists of a network of distributed, high performance digital signal processors, an associated front end analog signal input subsystem, and a set of graphical user interfaces.



Processing Network

The processing network is constructed using an appropriate number of Texas Instruments TMS320C40 and TMS320C44 digital signal processors (the number is dependent on system size and bandwidth). The processors are connected in an application-specific network using the intrinsic communication ports of the 320C4x. Processor memory is distributed (processors do not share memory), allowing the approach to scale up to very large systems. (This scalability is a result of the aggregate system bandwidth and computational capability scaling linearly with the number of processors.) Processors are grouped into sets, one set for each analog front-end set. These sets are responsible for performing most of the real-time analysis of the input signals. Processing of signals is gapless, i.e. results are based upon every sample. No samples are discarded. These sets of processors are arranged in a pipeline. Data for display is transmitted via this pipeline, as are commands from the user that control the behavior of the processors.

User Interface Hardware

The final major component of the CADDMAS is the user interface. The user interface is a GUI-based subsystem responsible for displaying real-time results. The UI is constructed using a high-performance PC with accelerated graphics. The accelerated graphics allows the system to display approximately 40-60 window updates per second. A window, or the entire screen can be printed at any time, with minimal impact on the processing of the system. Printers are attached via standard Centronics ports. The CADDMAS can support a variety of HP laser printers (or any HP-PCL5 printer). A system can support from 1 to 8 user interfaces. Each operator interface is independent. Facilities within the UI allow control of any UI from any other UI, allowing one operator to control multiple display screens.

CADDMAS User Interface Overview

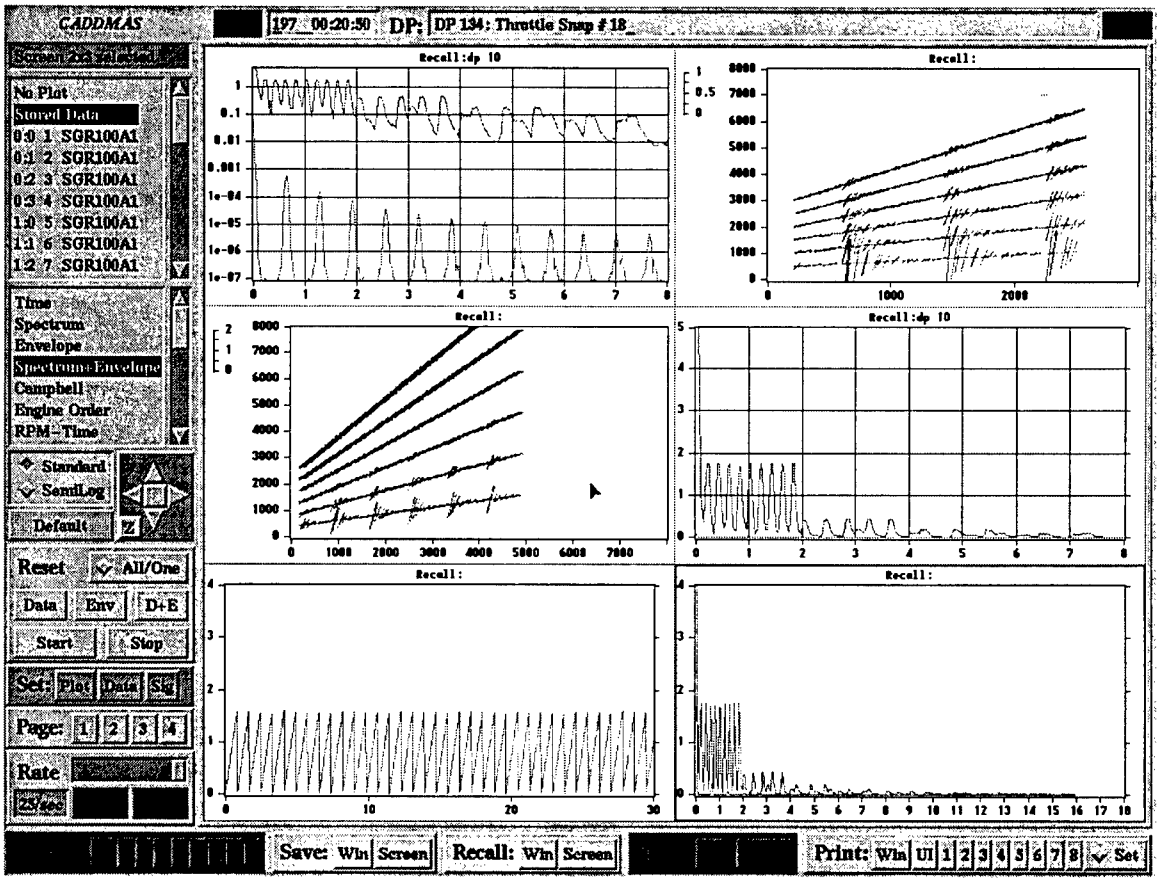
The CADDMAS user interface serves two functions:

- **High-Speed Display** of data from the parallel processor: The central section of the screen is dedicated to display. This area can be subdivided in a grid displaying as many as 25 plot windows. Plot windows can contain any combinations of channels and processing types. For each window, the following parameters are valid:
    - **Channel:** the physical channel in use.

- **Processing Type:** Time, Spectrum, Spectral Envelope, Spectral Waterfall, Campbell, Phase Campbell, Engine Order, Time-peak Vs RPM, TOPO, Frequency Tracking (Synchronous/Asynchronous), Autocorrelation, CrossCorrelation, Bicoherence, and special purpose as necessary.
- **Plot Formatting:** Axis scaling, log-scale, labels, etc.
- **Data Processing Parameters:** Campbell Min/Max RPM/Freq., Thresholds, Base Frequency, etc.
- **Update Rate:** the desired plot update frequency.

- **Interactive Control** of the computations being performed on the parallel processor. (Reset Envelope, Campbell, etc.)

The complete set of associated options and details of these functions are described in reference 1. The functions relevant to the machinery diagnostics application are described below



The following processing types are available:

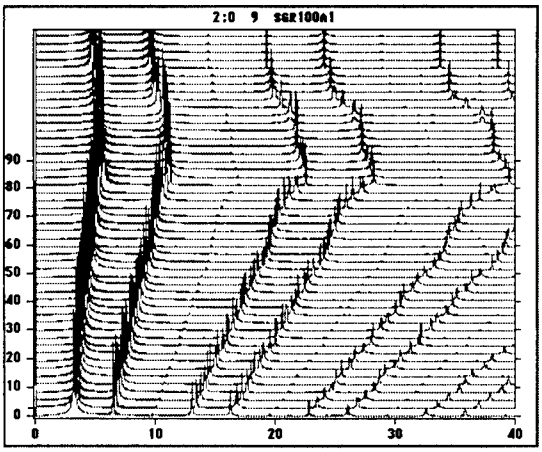
- **TIME:** The raw data input to the system, converted to engineering units. The Y scale is typically in KSI and the X scale is displayed in milliseconds.
- **SPECTRUM:** The magnitude spectrum, spectral energy vs. frequency. The spectrum is computed with a 1024, 2048, or 4096 point real input/complex output Fast Fourier Transform. The energy is computed as the absolute value of the complex spectrum and displayed in KSI peak-to-peak on the Y Axis. Frequency is displayed in KHz on the X axis. The spectrum can be displayed in waterfall format, with consecutive spectra displayed at fixed displacements along the Y axis.
- **SPECTRAL ENVELOPE:** A peak-hold for Spectral Magnitude signal. The data is displayed based upon ALL samples, using gapless processing. Maximum spectral energy at each frequency bin is retained until a user resets the envelope.
- **SPECTRUM/ENVELOPE:** The spectrum and spectral envelope data streams are displayed concurrently on the same plot.
- **CAMPBELL:** Spectral Peaks collected based upon RPM, frequency, amplitude, and the user-defined Campbell parameters (to be described later). The data collected for the Campbell is gapless.
- **ENGINE ORDER:** This process collects spectral magnitude at multiple of the engine rpm. The peak stress at a range of RPM is collected as a function of RPM. The X axis represents rpm while the Y axis is spectral energy in KSI. Up to 4 engine orders can be tracked at any one time for each signal.
- **RPM-TIME:** This process collects peak stress as a function of RPM (or other parameter), The X axis is RPM while the Y axis is absolute stress.
- **CROSSCORRELATION:** This process computes the CrossCorrelation of any two signals within the system. If the primary and secondary signals are the same, the computation is an Autocorrelation. The X axis is time shift while the Y axis is the correlation at that shift. The pairs of signals to correlate are selected in the **SETUP SIGNAL** screen described below.
- **TOPO:** This process is based on a NASA algorithm. The TOPO is used to examine the trajectories of spectral components, in both high and low noise environments. The processing is as follows: Spectra are averaged for a user defined period. After the averaging period, a noise floor is computed using a "rainfall algorithm". The noise floor is subtracted from the signal. The peaks are located and normalized relative to the largest peak. For each peak above a user-defined threshold, a horizontal line is drawn on the plot. The length of the horizontal line is relative to the magnitude of the peak. The X axis of the plot represents frequency in KHz, the Y Axis is time, and the peak width is relative spectral intensity above the noise floor.
- **BICOHERENCE:** The Bicoherence computes the second order cumulant of a signal. The purpose of this plot is to compute the strength of phase correlation between two discrete frequencies, or frequencies that are multiples of RPM. The basic computation is normalized to a 0-1 range.

- **TRACKING:** This plot allows for the tracking of synchronous and asynchronous frequencies. The process captures frequency peaks within a band centered around a multiple of RPM (or other parameter). The width of the frequency peak search is user defined, as is the RPM multiplier. Up to 8 different multipliers can be specified for each signal.

Description of Algorithms

The algorithms which are applicable to the machinery diagnostics and health monitoring application are described in the following sections:

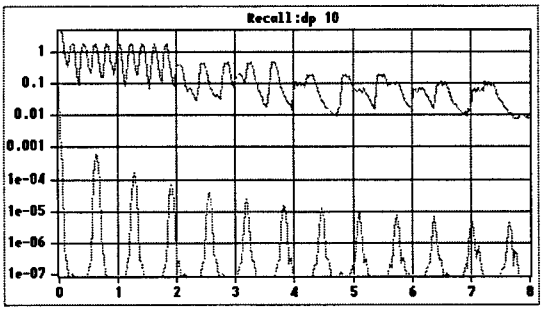
Spectrum Plots



The power spectrum of the signal is computed using 32 bit floating point arithmetic with the EU-converted time-domain data as input using a real-input, complex output FFT. A user-selectable window function is applied to the input block before FFT computation. While any window function can be added into the system, the currently supported functions are: **Rectangular, 4 point Blackman-Harris, Hamming, Hanning, and Gaussian**. FFT block size can be any valid block size (1024, 2048, or 4096). The power spectral density is computed by taking the absolute value of the complex FFT output.

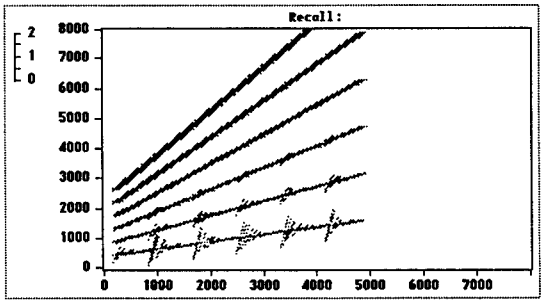
The power spectral density can be plotted alone, in normal or semilog-scale. The waterfall option allows multiple spectral lines to be displayed in succession (see above figure) for visualizing frequency trajectories.

Spectral Envelope Plots



The results of every spectral magnitude are incorporated into the envelope. The envelope is computed by taking the maximum of the historical envelope with the current spectral magnitude. This function is computed for each point in the spectrum and stored in the historical envelope. The envelope is accumulated continuously. The historical envelope can be reset by the user with a **RESET-ENVELOPE** command (see **RESET** above).

Campbell Plots



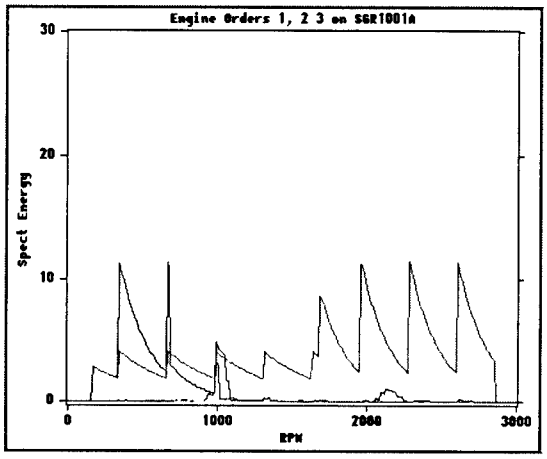
The Campbell plot can present a historical view of an engine maneuver, occurring over several minutes. The process to compute the Campbell is as follows:

1. The spectral peaks are located in the spectral magnitude data.
2. The peaks are refined with a parabolic interpolation process to estimate spectral energy and center frequency.
3. The Campbell Threshold is applied. Any peaks below the threshold are discarded.
4. For each peak, the associated rectangular bin is computed in the Campbell plot space. This bin is defined as a equal-spaced grid from Min to Max Parameter and Frequency. Within each bin, only the highest peak magnitude is retained and plotted.

5. The peaks are accumulated in sets of 2000 points. Each set is color-coded: Green, Purple, and Yellow, to let the user know when the end of Campbell collection memory is approaching.

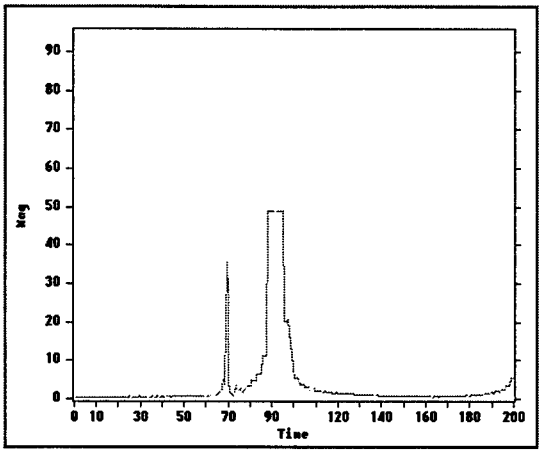
Proper setting of the threshold and collection range are important to ensure that the database size is sufficient to cover the entire Datapoint time.

Engine Order Plots



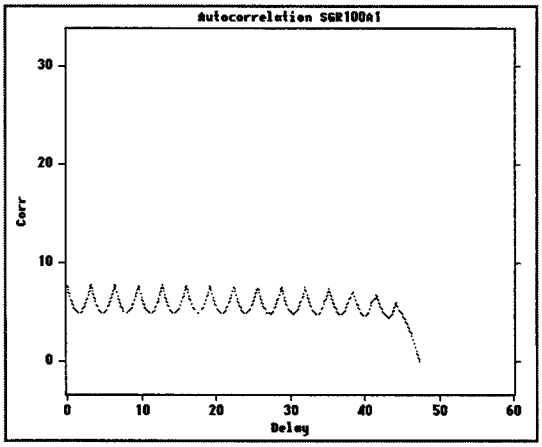
Engine order plots show the magnitude of the spectrum at user-specified multiples of the engine rotation frequency as a function of engine RPM. They are useful for determining relationships between forcing functions and excitations within the engine. For each spectral packet, an associated RPM is measured. The RPM frequency is multiplied by the user-specified EO Line parameter. The resultant frequency is used to index the spectral magnitude and retrieve the energy at that frequency. This energy is entered into the EO plot database at the current RPM if it is larger than the existing value. The resulting database is plotted, using RPM for the X axis and measured peak energy on the Y axis. The database can be reset under user control (see **RESET** above).

RPM-Time Plots



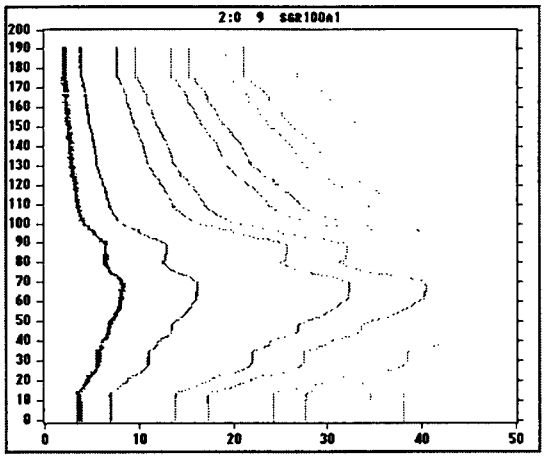
The **RPM-TIME** plot displays the peak stress experienced at a sensor as a function of the engine RPM. To compute the data, the peak of the time-domain signal is located, and interpolated to increase accuracy. The **RPM-TIME** database entry for the current RPM is replaced if it is larger than the existing value. The database can be reset under user control (see **RESET** above).

Auto/Crosscorrelation Plots



The CrossCorrelation function is useful for estimating the delay of corresponding signal phenomena. The CrossCorrelation is computed using the standard convolution-style algorithm. Any two signals within the system can be correlated, including signals with themselves. The pairing of signals for the XCORR operation is done through the SETUP-SIGNAL screen.

TOPO Plot



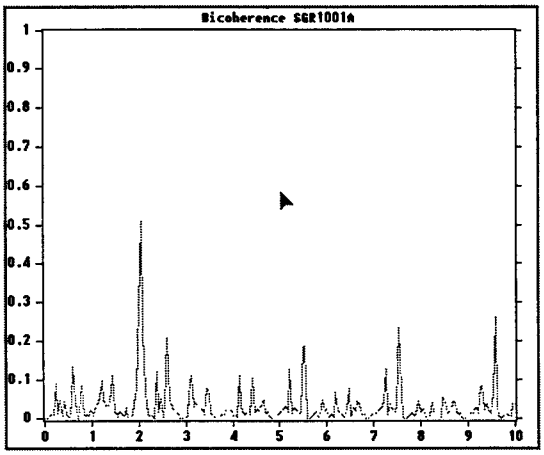
The **TOPO** plot is a NASA-developed algorithm that provides a method for visualizing trends in spectra. The basic organization is as follows:

- The X axis represents frequency. Each point in the plot represents a noise-reduced peak in the spectral magnitude data.
- The relative magnitude of the peak is represented with a horizontal bar: the length increases with peak magnitude.
- The lines in the plot are shifted up along the Y axis as a function of time.

The result is similar to looking down on a spectrogram, with the bar widths representing peak amplitude. Several enhancements are added to the processing to reduce noise in the plot and to deal with wide variations in spectral amplitudes:

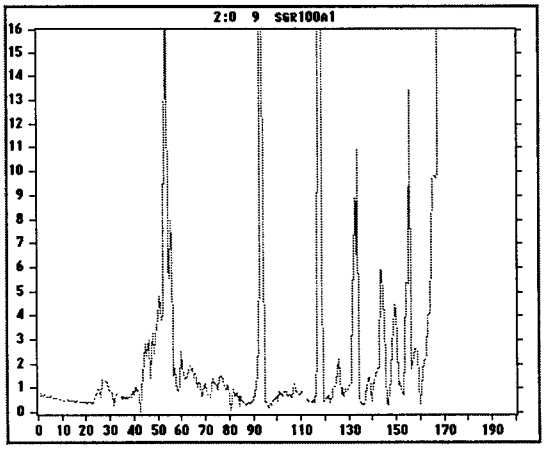
- Spectrums are averaged as defined by the user, (see PARAMETER SETUP) to reduce noise.
- A noise floor is computed. The floor follows the valleys in the spectral magnitude, with a special "Rainfall" algorithm to eliminate the effect of noise on the side of a large peak. Small dips are effectively ignored
- The computed noise floor is subtracted from the averaged spectrum and a the maximum value of the resulting signal is located.
- Peaks are located in the signal and added to the TOPO database.

Bicoherence Plots



The Bicoherence plot is a second order cumulant of a signal. Its primary function is to determine the phase correlation of two spectral lines in a signal. A high correlation indicates that the secondary signal is a result of excitation by the primary signal. A typical example would be to determine if a vibration on a blade was related to an engine shaft through a gearbox ratio. The basic algorithm computes a single line of the Bicoherence function, keeping the original signal and the signal shifted by the reference frequency constant and sweeping a copy of the original. The resultant signal is normalized and averaged a user-specified number of times for noise reduction.

Tracking Plots



The tracking plots allow tracking of individual spectral components over a period of time. The algorithm records the maximum spectral energy within a window around a computed base frequency. The base

frequency is either determined by a multiple of the RPM for synchronous tracking, or the base frequency can be retained from the previous data packet, for anomalous frequency tracking(anomalous tracking is not yet in the production version). The search window is determined by multiplying the base key phasor frequency by the Key Phasor Multiplier. The largest peak within the region BASE-WINDOW to BASE+WINDOW will go into the plot.

Future Work

The availability schedule of the telemetry module precluded use of the system in an application within the operational environment. The CADDMAS processing system has, however, been used in ongoing NASA experiments in a turbopump research facility and at NASA Stennis Flight Test Center as a red-line system for the SSME, using direct-connect A/D's. In this work, the CADDMAS has proven very successful.

Several experiments are planned for the telemetry-equipped system:

1. Monitoring AEDC's large wind tunnel compressors. These large rotating fans are extremely expensive, and equipment failure can lead to unacceptable downtime.
2. Monitoring of helicopter rotors.

We anticipate that these applications will be worked in the early part of 1997.

References

"CADDMAS User's Manual", AEDC Technical Report, 1997

"Real-Time Condition Monitoring for Propulsion Systems through
High Performance Computing", T. Zoladz; T. Fiorucci; T. Bapty, Jen Jong
32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Lake Buena Vista, FL July 1-3, 1996

T. A. Bapty et al.: "Parallel Turbine Engine Instrumentation System," Proc. of Computing in Aerospace 9, pp. 434-440, San Diego, CA, 1993

Texas Instruments, "TMS320C40 User's Guide," Texas Instruments Incorporated, Houston, TX, 1992

Tibbals, T.F., Bapty, T. A., Abbott, B.A,: "CADDMAS: A Real-Time Parallel System for Dynamic Data Analysis", ASME 94-GT-194, International Gas Turbine and Aeroengine Congress, The Hauge, Netherlands, June 1994

T. A. Bapty, B. Abbott: "Real-Time Turbine Engine Data Visualization," Transputer Research and Applications, D. L. Fielding (ed.), IOS Press, 1990

Fatigue Crack Growth Rates in Naturally-Corroded Aircraft Aluminum

J. D. Baldwin
Assistant Professor
School of Aerospace & Mechanical Engineering

GraduatePaul Benjamin
Assistant Professor

University of Oklahoma
Norman, OK 73019

Final Report for:
Summer Research Extension Program

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Oklahoma City Air Logistic Center

February 1997

FATIGUE CRACK GROWTH RATES IN NATURALLY-CORRODED AIRCRAFT ALUMINUM

J.D. Baldwin
Assistant Professor
School of Aerospace & Mechanical Engineering
University of Oklahoma

Abstract

The focus of this project was to explore the impact of corrosion damage on aircraft fuselage structures from both the material characterization and structural response standpoints. In terms of material characterization, statistical analyses were conducted to establish the correlation between degree of corrosion damage (i.e., depth of attack) and accelerated fatigue cracking behavior. The preliminary analysis results indicate that accelerated crack growth rates scale linearly with depth of attack for 7000-series aluminum alloys. Structural analysis results are expected to give the first indications of the effect of corrosion damage on larger structural elements. These analyses are being conducted using finite element structural analysis and fracture mechanics codes.

FATIGUE CRACK GROWTH RATES IN NATURALLY-CORRODED AIRCRAFT ALUMINUM

J.D. Baldwin

Introduction

This is a report of the 1995 AFOSR Summer Research Extension Program effort to quantify fatigue crack growth rates in naturally-corroded aircraft aluminum alloys. The project was monitored by Mr. Don Nieser, OC-ALC/LACRA at Tinker AFB, Oklahoma. The original scope of the investigation was to conduct fatigue crack growth rate tests on specimens harvested from retired USAF C/KC-135 fuselage and wing skins that had been damaged by corrosion while in service. The test specimens were to have been provided by OC-ALC/LACRA for testing. When it became apparent that the specimens could not be provided by that organization, a revised statement of work was submitted and approved in January 1997. The revised scope of investigation included conducting analytical studies of corroded material fatigue crack growth rates.

Discussion of Problem

It has been estimated that, service-wide, the U.S. Air Force spends $1 billion per year to repair corrosion damage on aircraft. The USAF contingent of C/KC-135 aircraft is an aging fleet, the youngest aircraft having been delivered in 1965, and currently there is no plan to replace the fleet in the near future. As these aircraft age, they become more susceptible to corrosion damage, especially in the fuselage lap joints and wing skins around fastener holes. At this time, however, very little data exists on the reduction in strength of a metal that has lost material to corrosion. Because the effect of corrosion on structural strength has not been allowed for in the design of these aircraft, current inspection intervals may be too long to guarantee structural safety.

The C/KC-135 fleet spends most of its time on the ground exposed to the spectrum of atmospheric contaminants that promote corrosion of the structure. During the ground time, the airframe is essentially unloaded. Fully loaded conditions, on the other hand, occur during the relatively short periods when the aircraft is in flight. We assume that the flight environment is relatively unaggressive with respect to corrosion of the aluminum when compared with the ground environment. Given that these aircraft missions involve primarily high altitude flight in rarefied atmospheres, this seems to be a reasonable assumption. It should be noted that, as important as atmospheric contaminants are to corrosion of aircraft structures, there is still insufficient data to characterize the environment at all locations and altitudes around the world [1]. Because of these loading characteristics, it will be assumed that the loading cycles occur such that fatigue cracks will nucleate and grow, in a relatively unaggressive environment, in metal that has already experienced corrosion and a corrosive environment on the ground. This assumption will allow us to examine the role of prior corrosion on crack propagation without addressing the specific nature of the corrosive environment and corrosion growth. Ideally, we would be able to create an appropriate corrosive atmosphere in the laboratory, but such an environment has not been defined for military aircraft.

It is interesting to note that essentially all of the published work in corrosion fatigue has focused on the behavior of new (i.e., noncorroded) materials in various corrosive environments [2]. Although the existing literature provides valuable insight into the corrosion fatigue behavior of metals, it fails to address the metal's response to loading after corrosion has already occurred. Only recently have experiments designed to quantify the fatigue response of corroded metal begun to appear [3,4]. The C/KC-135 Corrosion Fatigue test program [5] was designed to collect some basic fatigue crack growth rate data for pre-corroded materials and compare it with data for baseline,

noncorroded samples of the same materials.

The primary hypothesis of the original investigation was that by examining in the laboratory the fatigue response of pre-corroded material in relatively unaggressive environments (specifically dry air and moist air), we can model the behavior of actual structural materials in flight. Note that this environmental assumption is probably not valid for aircraft spending their flight time in marine environments where a salt spray atmosphere may be more appropriate.

During the summer of 1995, the Principal Investigator worked in the C/KC-135 Corrosion program at Tinker AFB, Oklahoma. During that time, fatigue crack growth rate testing was carried out on aircraft structural aluminum alloys corroded artificially, i.e., in the laboratory. The results of that effort [6], while not conclusive, suggested that fatigue cracks were accelerated by a factor of as much as two over noncorroded material when the corrosion had removed 3-5% of the specimen thickness.

The goal of this project was to perform tests to quantify the fatigue crack growth rate in naturally-corroded aircraft aluminum. This basic material response data, when compared with data for noncorroded material, will provide an estimate of the magnitude of fatigue damage acceleration due to corrosion. When compared with artificially grown corrosion [6], the natural corrosion data will provide an indication of the suitability of the artificially-corroded material data in a model of actual structural components. Once a suitable body of data is available, it may then be integrated into existing structural integrity program procedures.

When the original project was developed, it included a promise that the C/KC-135 Engineering Group at Tinker AFB, Oklahoma would provide machined test specimens for testing. These specimens were to be the central focus of the investigation. In mid-October 1996, the

Principal Investigator was informed that the specimens would not be available in a time frame to allow reasonable completion of the original statement of work. While waiting for the specimens we conducted statistical analyses of the fatigue crack growth rate data collected in the USAF C/KC-135 Corrosion Fatigue Program; the results of this effort are summarized below. To reflect the project's transition from an experimental to analytical focus, the following revised statement of work was formulated with the agreement of Mr. Don Nieser, OC-ALC/LACRA, the PI's Laboratory Focal Point on this project.

Recent research has shown [7] that the increase in fatigue crack growth rate (FCGR) in aluminum damaged by prior corrosion appears to scale with the amount of material lost. If there is only a single crack in a structure, or if there are multiple cracks that do not interact, the reduction in structural life in the presence of corrosion-damaged material will scale directly with the level of corrosion-induced thickness loss. In the case of multi-site damage where interacting cracks are anticipated, however, the increased FCGR's at the cracks are expected to have a synergistic effect on the structural integrity degradation. In this revised project, the overarching goal was to explore the impact of prior corrosion on the behavior of riveted lap joints containing multi-site damage. This investigation focused on the behavior of fatigue cracks growing from the rivet holes into material damaged by a known amount of corrosion damage; we anticipated that the corrosion severity could be "graded" throughout the joint to simulate the damage observed in actual fuselage teardowns, where the thickness loss is not uniform throughout the joint.

The importance of this effort was that, by using corrosion-based material properties in the MSD/WFD analyses, we would begin to explore the impact of corrosion damage on the structural level. Up to this point, corrosion has been considered in coupon testing, but no published studies

exist combining corrosion and MSD into a single analysis. Today no data exists on this behavior. Fundamental studies like this will yield valuable information for aircraft owners/operators on the relative impact of various aspects of corrosion damage on the continued durability of their airframes.

The goals of this investigation were to explore the impact of corrosion on airframe structural integrity by studying the anticipated synergistic effect of corrosion on crack growth and link up observed in riveted lap. As noted above, this study was to be analytical in nature using available mechanical models and state of the art computer codes. This task addressed the classical fuselage lap joint multi-site damage model, but expanded the analysis to include the accelerated fatigue crack growth rates observed in material damaged by prior corrosion [7]. The geometry studied was based on a typical three row rivet pattern as illustrated in Figure 1. The rivet spacings were taken to be 1 inch on centers (transverse and longitudinal) and various size cracks were assumed to have nucleated from the rivet holes along the top row of rivets. Using the NASA *FADD* program [8], the stress intensity factors at each crack tip can be computed for general in-plane loading combinations, e.g., Mode I and Mode II. Fatigue crack growth rate date (i.e., da/dN versus ΔK) derived from the USAF Round Robin Corrosion Fatigue tests [7] will be used along with the computed K's to estimate the number of load cycles (or cabin pressure cycles) required to drive the cracks to link up. The plastic ligament criterion [9], where cracks are presumed to join when the ligament between them is fully plastic, will be used as a first approximation. Although very little data is available on the effect of prior corrosion on fracture toughness [10], in this investigation, we will assume that fracture toughness scales linearly with thickness loss just like FCGR. By introducing the modified crack growth rate, defined as

$$(da/dN)_{corr} = \frac{t_{bl}}{t_{corr}} (da/dN)_{bl} \quad (1)$$

we could simulate degrees of corrosion-induced thickness loss and study that factor's impact on the deterioration of the structure. In Equation 1, t is the nominal sheet thickness and subscripts "bl" and "corr" represent the baseline (noncorroded) and corroded material conditions, respectively. Note that Equation 1 reflects the inverse relationship between sheet thickness and crack growth rate. By conducting parametric studies on various crack configurations and comparing with previous computational [11] and experimental studies [12-14] using the same and similar configurations, we will be able to establish bounds on the reduction in time to loss of residual strength due to corrosion damage in the joint. Also, using the corrosion-modified fracture toughness value, we can compute the residual stress for a MSD/WFD panel and compare with the noncorroded analogous case.

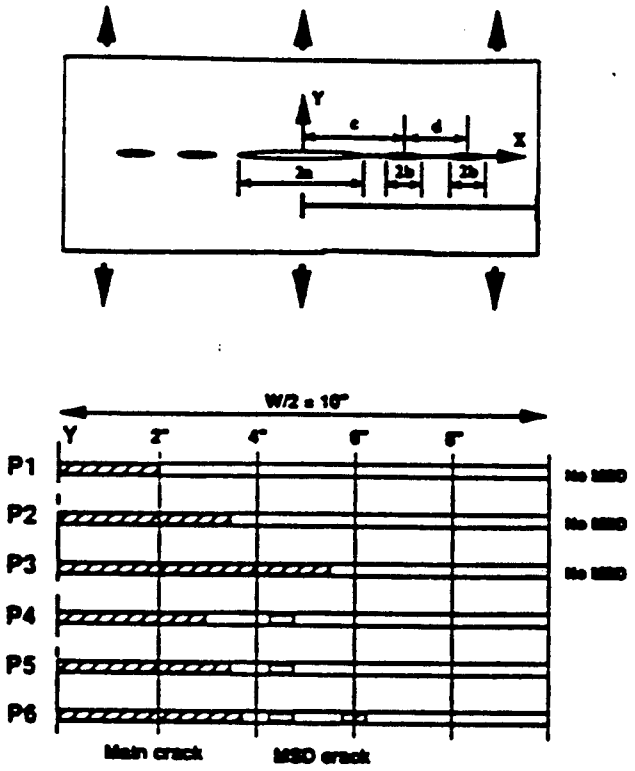


Figure 1: Schematic Description of MSD/WFD Models [11]

Results

To address the statistical variation in fatigue crack growth rate (FCGR) data and its influence on structural integrity, we examined test data for forty 7075-T6 specimens collected during the C/KC-135 Corrosion Fatigue Program. Of these specimens, twenty four were in the baseline (or noncorroded) condition and sixteen had been damaged by corrosion to a depth estimated to be 16% of the initial thickness. In the study of the statistical validity of a linear relation between depth of corrosion attack and crack acceleration, we restricted our attention to a simple fourth-order polynomial model, given by

$$\log da/dN = k_0 + k_1(\log \Delta K) + k_2(\log \Delta K)^2 + k_3(\log \Delta K)^3 + k_4(\log \Delta K)^4 \quad (2)$$

This equation form does a good job of capturing the sigmoidal shape of the FCGR data (something the log-linear Paris equation cannot do) and has the computational advantage of being a linear regression calculation, as opposed to the more problematic nonlinear regression required for other proposed relationships. To illustrate the adequacy of the quartic polynomial for describing *da/dN* versus ΔK data, the average coefficients of determination for each of four test series taken from the C/KC-135 Corrosion Fatigue Program are summarized in Table 1. These values indicate that the model is a reasonable reflection of the data in the intervals where the data exist. Whereas the curve fits can be expected to provide good interpolation *between* data points, they clearly cannot be used to extrapolate beyond the domain of the data.

For this analysis, the quartic curve fits were used to estimate the *da/dN* values for each experiment (specimen) at $\Delta K = 2, 4, 6, \ldots 20$ ksi·√inch. Because the starting stress intensity range was 3-5 ksi·√inch in these experiments, the $\Delta K = 2$ ksi·√inch values were not used in the statistical analysis. Also, at the higher ΔK ranges, the data was censored to remove points representing invalid plastic zone sizes according to ASTM E 647[15]. With two experimental data sets (baseline and corroded) established at a given value of ΔK, the issue of whether the data sets can be assumed equivalent, i.e., representative of a single population, can be addressed. The following statistical hypothesis was posed:

H0: At the $\alpha = 0.01$ significance level, the mean baseline material *da/dN* is equal to the corroded material mean *da/dN* at a given ΔK

To explore this question, an unpaired Student *t*-test was run on each ΔK group. If the null hypothesis was rejected based on the data, we concluded that there was a statistically-significant difference between the baseline and corroded material crack growth rates. If we fail to reject the null

hypothesis, we must accept the possibility that there is no difference.

For purposes of the statistical analysis, the four 7075-T6 test series summarized in Table 2 were considered separately; no attempt was made at comparing differences due to stress ratio or relative humidity. The ranges of ΔK where statistically-significant differences in the crack growth rates were observed were clustered at the lower ΔK values, especially where the curve turns downward toward vertical. Also, we observed that the tendency for differences in *da/dN* to disappear at higher ΔK levels was confirmed by the statistical analysis. Therefore, based on the original specimen thickness, we conclude that these data show a statistically-significant difference between baseline and corroded material, primarily at low to moderate ΔK levels.

It was noted above that a second reason for developing and implementing the current statistical analysis of FCGR data was to address the validity of modeling corrosion as mechanical damage. Doerfler, et al. [16] originally proposed this simplification without supporting data. Scheuring and Grandt [3], based on a subjective analysis, concluded that, for "light" corrosion, differences in FCGR were accounted for by making an appropriate thickness correction. To explore this issue further, another set of statistical analysis runs were made on the 7075-T6 data. A post fracture microscopic examination of the two shortest-life specimens revealed that, on the fracture surface, the thickness lost to corrosion was between 12% and 16% of the nominal thickness. These thickness reduction figures represent average values on the specimen cross section. More detailed analysis may be able to give more localized thickness data, thus allowing improved estimates of the instantaneous value of ΔK at various crack positions, but it is felt that this added sophistication will not affect the engineering accuracy of the material characterization. Using the 16% thickness reduction figure as representative (and recalling that such values were not available for all the

specimens), each specimen thickness was reduced by 16% and modified ΔK's were computed for each data point. The resulting *da/dN* versus ΔK are plotted in Figure 2. In this Figure, it can be seen that the corroded material data now lies on top of the baseline data throughout a much larger range of ΔK. In fact, as Table 5 shows, the regions of statistically-significant differences for the reduced thickness case effectively disappear. The exception to this trend is in the $R = 0.05$, < 15% R.H. data set, where the corroded material data are seen to lie below the baseline data. In this case, the thickness correction has resulted in corroded material FCGR's that, as shown in Table 5, are significantly slower than the baseline material. This result is unexpected and is probably not correct physically. Under no circumstances would we expect corrosion damage to retard crack growth. This test series suggests that the thickness correction should be fully applied at higher ΔK levels. In spite of this unusual occurrence, the other test series are found to confirm the assumption that thickness corrections are sufficient to explain FCGR increases in the presence of corrosion damage.

Conclusions

The results of this project shed new light on the behavior of corroded aircraft fuselage structure. The primary finding is that the fatigue crack growth behavior in corroded material (particularly 700 series aluminum alloys) scales linearly with the depth of corrosion attack. This result is currently being incorporated into finite element models of fuselage skin and structure to estimate the degradation of structural integrity due to corrosion. The exploration of multiple site cracking in corrosion-damaged structure is continuing, as is the material characterization aimed at providing vital material performance data for aging aircraft structural analysis.

References

[1] Wallace, W., D.W. Hoeppner and P.V. Kandachar, *AGARD Corrosion Handbook, Volume 1 - Aircraft Corrosion: Causes and Case Histories*, Neuilly Sur Seine, France: Advisory Group for Aerospace Research & Development, July 1985.

[2] Gangloff, R.P., *Corrosion Fatigue Crack Propagation in Metals*, NASA CR 4301, Washington D.C.: National Aeronautics and Space Administration, 1990.

[3] Scheuring, J.N. and A.F. Grandt, "An Evaluation of Fatigue Properties of Aging Aircraft Materials," presented at the 1994 USAF Structural Integrity Program Conference, San Antonio TX.

[4] Koch, G. H., "The Effects of Pre-Existing Corrosion on the Fatigue Cracking Behavior of Aircraft Aluminum Alloys", presented at the USAF Third Aging Aircraft Conference, Wright-Patterson AFB OH, 1995.

[5] *Integrated C/KC-135 Corrosion Program Round Robin Testing: Pre-Corroded Fatigue Crack Growth Rate Test Plan*, Wichita KS: Boeing Defense & Space Group, Product Support Division, March 1995.

[6] Baldwin, J.D., T*he Effect of Corrosion on Fatigue Crack Growth Rates in Aircraft Structural Aluminum Alloys*, Final Report for the 1995 AFOSR Summer Faculty Research Program, September 1995.

[7] J.D. Baldwin, 1996, *Statistical Analysis of Fatigue Crack Growth Rate Data for 7075-T6 Aluminum Damaged by Prior Corrosion*, Final Report, AFOSR Summer Faculty Research Program.

[8] C. Chang, L. Xiao, M. Mear, V. Kale, K. Madhusudana, J.C. Newman, Jr. and C.E. Harris, 1996, *FADD: Fracture Analysis by Distributed Dislocations - Version 1.0 User's Guide for Workstations*, NASA.

[9] D.Y. Jeong and J.C. Brewer, 1995, "On the Linkup of Multiple Cracks," *Engineering Fracture Mechanics*, 51 (2), 233-238.

[10] J.P. Chubb, T.A. Morad, B.S. Hockenhull and J.W. Bristow, 1995, "The Effect of Exfoliation Corrosion on the Fracture and Fatigue Behavior of 7075-T6 Aluminum," *International Journal of Fatigue*, 17(1), 49-54.

[11] C.R. Pyo, H. Okada, L. Wang, F.W. Brust and S.N. Atluri, 1995, "Residual Strength Prediction for Aircraft Panels with Multiple Site Damage (MSD) Using the Elastic Plastic Finite Alternating Method (EPFEAM)," in *Structural Integrity of Aging Aircraft*, C.I. Chang and C.T. Sun, eds., ASME, 73-80.

[12] D. Broek, 1993, *The Effects of Multi-Site-Damage on the Arrest Capability of Aircraft Fuselage Structures*, TR 9302, FractuREsearch Corp.

[13] L. Molent and R. Jones, 1993, "Crack Growth and Repair of Multi-Site Damage of Fuselage Lap Joints," *Engineering Fracture Mechanics*, 44 (4), 627-637.

[14] G. Samavedam and D. Hoadley, 1994, *Fracture and Fatigue Strength Evaluation of Multiple Site Damaged Aircraft Fuselages - Curved Panel Testing and Analysis*, Final Report, DOT/FAA/CT-94/10, DOT-VNTSC-FAA-93-8.

[15] ASTM E-647, *Test Method for Measurement of Fatigue Crack Growth Rates*, Philadelphia PA: American Society for Testing and Materials, 1993.

[16] M.T. Doerfler, A.F. Grandt, R.J. Bucci and M. Kulak, "A Fracture Mechanics Based Approach for Quantifying Corrosion Damage," *Proceedings of the Tri-Service Conference on Corrosion*, Orlando Florida, 433-444, 1994.

Table 1: Mean Coefficients of Determination From Quartic Curve Fits of *da/dN* vs. ΔK Data

	Mean Coefficient of Determination, r^2		
Test Series	Baseline	No Thickness Corr.	16% Thickness Corr.
$R = 0.05$, < 15% R.H.	0.9900	0.9355	0.9355
$R = 0.05$, > 85% R.H.	0.9781	0.9637	0.9637
$R = 0.50$, < 15% R.H.	0.9772	0.9714	0.9719
$R = 0.50$, > 85% R.H.	0.9806	0.9620	0.9612

Table 2: Accomplished Test Matrix

	Baseline (Non-corroded)				Corroded				
	$R = 0.05$		$R = 0.50$		$R = 0.05$		$R = 0.50$		
Material	<15%	>85%	<15%	>85%	<15%	>85%	<15%	>85%	Total
7075-T6	6	6	6	6	4	4	4	4	40

PERFORMANCE ANALYSIS OF PROBABILISTIC POTENTIAL FUNCTION NEURAL NETWORK

Gursel Serpen
Assistant Professor
Department of Electrical Engineering and Computer Science

University of Toledo
2801 West Bancroft Street
Toledo, OH 43606

Final Report for:
Summer Faculty Research Program
Ogden Air Logistics Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Ogden Air Logistics Center

February 1997

PERFORMANCE ANALYSIS OF PROBABILISTIC POTENTIAL FUNCTION NEURAL NETWORK

Gursel Serpen
Assistant Professor
Department of Electrical Engineering and Computer Science
University of Toledo

Abstract

Simulation analysis of newly proposed Probabilistic Potential Function Neural Network classifier algorithm on a set of benchmark problems was performed. Benchmark problems included IRIS, Sonar, Vowel, Two-Spiral, Wisconsin Breast Cancer Disease, Cleveland Heart Disease and Thyroid Gland Disease data sets. The performance of Probabilistic Potential Function Neural Network algorithm on these benchmark problems was compared to the performance of other important neural network classification algorithms, which included Multi-Layer Perceptron Network, Learning Vector Quantizer Network, Radial Basis Function Network, and Probabilistic Neural Network. Specially, classification performance of each algorithm was studied. Simulation results indicate that the Probabilistic Potential Function Neural Network offers fast training cycle, implements on-line and incremental learning, offers robust performance characteristics against variations in values of heuristically determined parameters, requires minimal computational resources, and topologically adapts its network structure to the requirements of the classification problem. As a result, the Probabilistic Potential Function Neural Network is suitable for classification problems which require real-time solution and does not initially come with large training data sets as characterized by the Neural Radiant Energy Detection System developed in Ogden Air Logistics Center.

PERFORMANCE ANALYSIS OF PROBABILISTIC POTENTIAL FUNCTION NEURAL NETWORK

Gursel Serpen

1. Introduction

A thermal (infra-red) imaging system, Neural Radiant Energy Detection System (NREDS) has been successfully applied to diagnose faults in electronic circuit cards [5][6]. The NREDS classification algorithm, a Bayesian classifier, performs a two-level procedure to isolate the faulty components on a given circuit card. Initially, a high level decision is made to classify a given card into one of two classes: the class of operational cards and the class of non-operational cards [6][7][8]. The input to the classifier is an $n \times 1$ vector components of which are the temperature rise rates associated with each of n components on the circuit card. The output of the classifier is the probability of the card under test belonging to the class of operational cards. Additionally, a confidence measure is computed.

The main underlying assumption in the current data model is that a uni-modal normal distribution can accurately model the true class density. The class Probability Density Function (PDF) will fail to adequately approximate the actual class PDF if pattern sets of a given class are clustered in separated and disconnected regions of the pattern space. As a result, the performance of the current classification algorithm of NREDS has been observed to deteriorate greatly for multi-modality data. The NREDS classification algorithm tends to cover regions in-between disjointed clusters of the same class.

Artificial Neural Networks (ANN) have been shown to perform satisfactorily for a number of challenging adaptive pattern recognition problems [1][2][3]. Ability to learn from examples and to generalize as well as an ability to form arbitrarily complex decision boundaries (an important feature for estimation of PDFs for multi-mode and disjoint classes) establish ANNs as a potentially very promising paradigm for the NREDS classification task [7][8]. ANNs also offer parallel computing and real-time operation capabilities which are highly desirable features for practical systems.

1.1 Survey of Literature

There are four important Artificial Neural Network (ANN) paradigms among others which have been extensively applied to pattern classification tasks successfully in the literature and are promising for many classification tasks [1][2][3][4]. These ANN paradigms include Multi-Layer Perceptron (MLP) network, Radial Basis Function (RBF) network, Learning Vector Quantization (LVQ) network and Probabilistic Neural Network (PNN). A brief overview of strengths and weaknesses of these neural paradigms are presented below.

RBF networks, also known as networks of locally-tuned processing units, are well known for their ease of use [9]. RBF neural network can be trained up to three orders of magnitude faster than MLP's for the same type of

problems. Initialization of the network requires clustering properties of the data to be analyzed and understood well, which is typically performed using an unsupervised learning algorithm like the *k*-means and is essential to determine the number of hidden layer nodes and their parameter settings. The training process requires matrix inversion operation to be performed to compute the weights from the hidden layer to the output layer.

Most algorithms used to train this type of RBF network, however, require a fixed architecture, in which the number of units in the hidden layer must be determined before the training starts [9]. Another algorithm based on RBF algorithm, RCE training algorithm, which was introduced by Reilly, Cooper and Elbaum [10] [14], and its probabilistic extension, the P-RCE algorithm, take advantage of a growing structure in which hidden units are only introduced when necessary. The nature of these algorithms allows training to reach stability much faster than is the case for gradient-descent based methods. Unfortunately, P-RCE networks do not adjust the standard deviation of their prototypes individually, using only one global value for this parameter. An improved algorithm, called Dynamic Decay Adjustment (DDA) algorithm, proposed by Berthold and Diamond [11], utilizes the constructive nature of the P-RCE algorithm together with independent adaptation of each prototype's decay factor. In addition, this radial adjustment is class dependent and distinguishes between different neighbors. It is shown that networks trained with the proposed algorithm perform substantially better than common RBF networks. On the other hand, a probabilistic extension for the DDA algorithm has been proposed by Berthold [12] [13], which uses the DDA to find conflict free areas and builds more appropriate PDFs inside each such zones on a data set which was generated using Gaussian distributions. It was demonstrated that this method builds almost optimal classifiers that compare very well with the theoretical Bayes classifier. It is shown, however, that the generalization capability of such networks does not compare favorably to the DDA itself. However, still, it is possible that a lack of understanding of the clustering properties of the data may cause an inappropriate network topology to be specified, which in turn will cause the performance to suffer significantly [15][16].

The LVQ methods are closely related to certain paradigms of self-organizing neural networks. LVQ networks can be trained very efficiently as compared to other neural paradigms, MLP and RBF networks [17]. Among adaptive LVQ methods those based on supervised learning have yielded very high pattern recognition accuracies. Their classification accuracy is at least as high as that of any other ANN algorithms. Due to the very simple computations thereby applied, their speed in learning as well as in classification can be significantly higher. Moreover, they are very easy to use. But while the classification accuracy of the LVQ algorithms has been demonstrated to be very close to the decision-theoretic Bayes limit even in difficult cases, nonetheless, some problems have remained; one of them is optimal initialization of the codebook vectors. This paradigm suffers significant performance degradation if the codebook vectors can not be initialized optimally, for which no well-defined procedure exists. The effect of initialization on the network performance gets worse if the class distributions are disjoint and maybe even intermingled [18][19][20].

PNN paradigm is formed by replacing the sigmoid activation function often used in neural networks with an exponential function [21]. PNN can compute nonlinear decision boundaries which approach the Bayes optimal.

A four-layer neural network of this type proposed can map any input pattern to any number of classifications. The decision boundaries can be modified in real-time using new data as they become available, and can be implemented using artificial hardware "neurons" that operate entirely in parallel. Provision is also made for estimating the probability and reliability of a classification as well as making the decision. It has a tremendous speed advantage over the backpropagation algorithm. An improved method, called a maximum likelihood method, is proposed by Streit and Luginbuhl [22]. It is employed for training PNN using a Gaussian kernel, or Parzen window. This proposed training algorithm enables general nonlinear discrimination and is a generalization of Fisher's method for linear discrimination. Important features of this maximum likelihood training for PNNs are: 1) it economizes the well-known Parzen window estimator while preserving feed-forward ANN architecture, 2) it utilizes class pooling to generalize classes represented by small training sets, 3) it gives smooth discriminant boundaries that often are "piece-wise flat" for statistical robustness, 4) it is very fast computationally compared to back-propagation, and 5) it is numerically stable. A potential problem with the PNN paradigm is that it requires a pattern layer node to be created and tuned to (weight vector of the newly created node is set to the training pattern) for each training pattern, which might result in very large node counts in the pattern layer for some realistic size problems.

PNN algorithm will perform satisfactorily given that the classes are separable and there exists class boundaries of the type defined by its potential functions [23][24]. Consistency of the training data set where training set is representative of the testing set and does not have conflicting information (wrong class assignment for a particular training pattern) will play an important role for the performance of the PNN, which is typical for any pattern classification algorithm. A number of theorems related to the convergence properties, the rate of convergence and conditions for termination of the algorithm are presented in [25].

The MLP with back-propagation is the best-known and most often used neural network paradigm today [26][27][28][29]. But it is difficult to use while exploring new preprocessing techniques and new data bases due to its exceedingly slow training. The lack of efficient techniques to determine the topology of the network for a given problem and the slow learning speed make this paradigm unsuitable for real-time implementations. It is well documented in the literature that the number of hidden layer nodes play a very important role in the ability of the network to partition the pattern space and currently there are no well-defined analytical procedures to specify the number of hidden layer nodes for a particular problem except in a number of limited cases.

Numerous attempts to improve the learning rate and learning performance of the backpropagation algorithm exist in the literature. One new learning algorithm, proposed by Fahlman [30], named "quickprop" is considerably faster than standard backpropagation. It was also pointed out that the quickprop appears to scale up very well as the problem size increases. In another case, a novel algorithm is presented which supplements the training phase in feedforward networks with various forms of information about desired learning properties [31][32][33]. This form is represented by conditions which must be satisfied in addition to the demand for minimization of the usual mean square error cost function. The purpose of these conditions is to improve convergence, learning speed, and generalization properties through prompt activation of the hidden units, optimal

alignment of successive weight vector offsets, elimination of excessive hidden nodes, and regulation of the magnitude of search steps in the weight space. It is found that its performance in terms of percentage of local minima, learning speed, and generalization is superior to the performance of the backpropagation algorithm.

All four ANN classifier algorithms, RBF, PNN, LVQ and MLP, lack one or more required features to satisfy the computational requirements of NREDS classification task. This, in turn, necessitated the custom design of a novel neural classifier algorithm, Probabilistic Potential Function Neural Network (PPFNN), which was proposed by Serpen, et.al. [34][35].

1.2 Probabilistic Potential Function Neural Network

In a typical stochastic pattern classification problem, noise or other real-life imperfections may cause the class sets to overlap in the pattern space and therefore making it impossible to assign a given pattern to a particular class with certainty. In this case, a probability value for class membership of a pattern can be computed to determine the class to which the pattern most likely belongs. The PPFNN has been designed to handle the requirements of a stochastic decision making problem.

The PPFNN algorithm offers the following theoretical promises:

1. PPFNN algorithm can be trained on line (fast learning speed) and classify in real-time even if implemented in software,
2. It can form classification boundaries which optimally separate the classes which are likely to be formed from a set of disconnected subclasses in the pattern space; the joint probability density function (PDF) of a particular class is likely to have many modes,
3. It does not require an initial guess for the network topology, rather topologically adapt to a particular instance of the classification problem at hand in a dynamic way as the training progresses,
4. It can discover clustering properties of training data and adapt to a minimal network topology in terms of needed computational resources,
5. It can implement incremental learning procedure and hence, does not disturb the previous state of the network but simply adds new computational resources to the existing network topology to learn the new training pattern, and
6. It can form optimal decision boundaries which approximate those of the theoretical Bayesian classifier.

1.3 Discussion of Bayesian Optimality of PPFNN Classification Boundaries

It is shown in the literature that the potential function method on which the PPFNN algorithm is based can form Bayesian optimal decision boundaries [36][37][38][39]. Additionally, convergence of the algorithm is proved: a theorem on the convergence (in the probabilistic sense) of the function constructed by the algorithm to the given degree of reliability is presented in references [38][39]. An algorithm is given based on the method of potential functions and permitting a function to be constructed which classifies the input patterns to a certain reliability. The

method of potential functions is used to construct the algorithm that makes it possible to use the results of random observations of a multi-input functional converter to restore its characteristic, which is assumed to be unknown. We now present a brief summary of relevant discussion in stated references.

> *Let there appear situations at the input to a machine each of which could be assigned to one of two classes A or B. It is assumed that for each of the situations there exist probabilities of belonging to the classes A and B, that in the learning process each situation is assigned to A or B with these probabilities. The set of all the situations which might appear at the input to the machine forms a space X.* $D_A(x)$ *and* $D_B(x)=1-D_A(x)$ *defined in all X, are termed the "reliabilities" that the point x belongs to the classes A or B. The problem consists of determining from the points appearing during the learning process and from the information supplied by the "teacher" as to the membership in A or B,* $D_A(x)$ *and* $D_B(x)$ *as functions prescribed over the entire space X. The problem can be solved on the assumption that* $D_A(x)$ *and* $D_B(x)$ *could be represented by a finite series expansion in some orthonormal system. In principle, the problem could also be solved by the method of maximum likelihood. However with the assumptions adopted as to the form of the reliability function, the equations arising in the method of maximum likelihood were practically not solvable (in particular due to the large number of parameters, the coefficients of the expansion, to be determined). This problem was solved by the use of a certain modification of the method of potential functions.*

2 Benchmark Problems and Neural Network Configurations

A thorough and comprehensive performance evaluation of PPFNN required an extensive set of benchmark problems to be employed. The set of benchmark problems included Two-Spiral, IRIS, Sonar, Vowel, Wisconsin Breast Cancer, Cleveland Heart Disease and Thyroid Gland Disease data sets [48][49]. A brief description of each benchmark problem and associated neural network configurations will be presented in the following sections. Additionally, more detailed information for the configuration of the neural network classifiers and values of parameters for all test cases are presented in Appendix I.

2.1 Two-Spiral Data Set

The task is to learn to discriminate between two sets of training points which lie on two distinct spirals in a two-dimensional plane [41]. These spirals coil three times around the origin and around one another. Each class has 96 points. There are a total of 192 input/output pairs in both training and test data sets: the training and test data set are the same. This is a two-class classification task. Training on the 192 I/O pairs continue until the learning system could produce the correct output for most of the inputs. LVQ, RBF, MLP, PNN and PPFNN neural networks were used for this classification task.

2.2 IRIS Data Set

The IRIS data set has 150 samples and 3 classes, each of them has 50 instances and each class refers to a type of IRIS plant [42][43]. They are IRIS Setosa, IRIS Versicolour and IRIS Virginica. One class is linearly separable from the other two; the latter two are not linearly separable from each other. The predicted attribute is the class of IRIS plant. There are four numeric input attributes (all in cm): sepal length, sepal width, petal length, and petal width.

All 150 instances in the data set were divided into five groups evenly. Four groups were used as training data and one group was employed as the test data for every experiment. A total of five simulation experiments were performed, so each group could be treated as test data. The average classification rates for both the training data and the test data were considered. The set of neural network algorithms used for this classification task included LVQ, RBF, MLP and PPFNN. Since this is a three-class classification task, PNN algorithm is not used here.

2.3 Sonar Data Set

This data set is used to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock [44][45]. This problem has two classes: mines and rocks. This leads to an output dimension of two. The dimension of input data is 60. There are a total of 208 samples [48][49], and each class has 104 samples. A total of 208 samples were divided into 13 groups randomly: each group had 16 samples. For each experiment, 12 groups were used as training data and the remaining group was used as test data. Thus, each sample could be used as test data. Simulation program was run 13 times. Average values of 13 runs were used as the performance measure. The set of neural network classification algorithms used for this classification task included LVQ, RBF, MLP, PNN and PPFNN.

2.4 Vowel Data Set

The data set is used to train a network for speaker independent recognition of the eleven steady state vowels of British English using a specified training set of derived log area ratios [40]. Vowels are classified correctly when the distance of the correct output to the actual output is the smallest among the distances from the actual output to all possible target outputs. The data needs to be classified into eleven classes. There are 990 samples, and they belong to 11 classes evenly. These eleven classes are hid, hId, hEd, hAd, hYd, had, hOd, hod, hUd, hud, and hed. Each class has 90 samples. The input dimension is 10 and the output dimension is 11. Ninety instances for each class were divided into two groups randomly: one group for training data and another for test data. The training data had 75 samples and the test data had 15 samples for each class. The program was run 6 times, so that each instance could be used as test data. The set of algorithms used for this classification task included LVQ, RBF, MLP and PPFNN. Note that PNN algorithm is not applicable here because there are eleven classes.

2.5 Wisconsin-Breast-Cancer Disease Data Set

The data set has 699 instances [10]. The number of attributes is 10 plus the class attribute. The first attribute is the sample code number (identification number, which we did not use in our simulation experiment). Attributes 2 through 10 have been used to represent instances. Attributes 2 through 10 are clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Each instance belongs to one of the two possible classes: benign or malignant. There are 16 instances that contain a single missing attribute value, denoted by "?". Resultantly, we used 683 instances for our classification task. There are 458 benign instances, and 241 malignant instances.

683 instances were divided into 11 groups. Each time 10 groups were used for training and one group for testing. A total of 11 experiments were conducted, each of which had 10 groups as training data and one group as test data. This led to each group to be treated as the test data. The average values of performance measures were used for comparison purposes. The set of neural algorithms employed for this data set included RBF, MLP, LVQ and PPFNN.

2.6 Cleveland-Heart Disease Data Set

This database contains 76 attributes, but all published experiments refer to using a subset of only 14 of them [46]. Attributes used in this simulation study include:

1) *age*: age in years.
2) *sex*: male or female.
3) *cp*: chest pain type [typical angina, atypical angina, non-anginal pain, asymptomatic].
4) *trestbps*: resting blood pressure (in mm Hg on admission to the hospital).
5) *chol*: serum cholestoral in mg/dl.
6) *fbs*: (fasting blood sugar > 120 mg/dl) [true, false].
7) restecg: resting electrocardiographic results [normal, having ST-T wave abnormality (T wave inversions and / or ST elevation or depression of > 0.05 mV), showing probable or definite left ventricular hypertrophy by Estes' criteria].
8) *thalach*: maximum heart rate achieved.
9) *exang*: exercise induced angina [yes, no].
10) *oldpeak*: ST depression induced by exercise relative to rest.
11) *slope*: the slope of the peak exercise ST segment [upsloping, flat, downsloping].
12) *ca*: number of major vessels (0-3) colored by flourosopy.
13) *thal*: [normal, fixed defect, reversable defect].
14) *num*: diagnosis of heart disease (angiographic disease status). [< 50% diameter narrowing, > 50% diameter narrowing].

303 instances were obtained from this data set. The goal is to discriminate the heart disease presence from absence. There are 164 normal instances and remaining 139 instances indicate the presence of heart disease. The data set was divided into 10 groups. For each simulation run, 9 groups were used for training and one group was used for testing. This arrangement led each group to be used as test data. The average values for performance criteria were employed for comparison purposes. The set of neural networks employed for the Cleveland Heart Disease data set included MLP, RBF, LVQ, PNN and PPFNN.

2.7 Thyroid-Gland Disease Data Set

There are 215 instances for this data set [40]. The number of attributes is 6. These attributes are:

1) class attribute [normal, hyper, hypo].
2) T3-resin uptake test (a percentage).
3) total serum thyroxin as measured by the isotopic displacement method.
4) total serum triiodothyronine as measured by radioimmuno assay.
5) basal thyroid-stimulating hormone (TSH) as measured by radioimmuno assay.
6) maximal absolute difference of TSH value after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value.

All attributes are continuous. Among all these 215 instances, 150 are normal instances, 35 are hyper instances and 30 are hypo instances. The data set was evenly divided into 5 groups. Each time 4 groups were used for training, and 1 group was used for testing. Average values of performance measures were employed in the comparisons. The neural networks used for the Thyroid gland disease data set included MLP, RBF, LVQ, and PPFNN.

2.8 Configuration of Neural Networks

MLP neural network configurations were determined using earlier work in the literature where applicable, Figure 2.1. Configuration of MLP algorithm for the Two-Spiral data set was modeled after the work by Lang and Witbrock [41]. The MLP networks for the IRIS, Sonar, Vowel and Wisconsin Breast Cancer data sets were configured as suggested by Gorman [44][45]. The MLP networks for Cleveland Heart Disease and Thyroid Gland Disease data sets were constructed following the work in [46]. Following observations hold for all MLP networks:

1) Nodes in hidden layers and output layer used sigmoid function and pure linear function, respectively, as their transfer functions.

2) Weights of networks were initialized to small random values uniformly distributed between -0.3 and 0.3. This was done to prevent the hidden units from acquiring identical weights during training.

3) Number of nodes in the input layer and output layer is equal to the dimensionality of the training patterns and number of classes, respectively.

	2-Spiral	IRIS	Vowel	Sonar	Wisconsin	Cleveland	Thyroid
Layers	4	4	3	3	3	4	4
Input Nodes	2	4	10	60	9	14	6
Hidden Nodes	10, 10	100, 100	24	24	24	10, 10	10, 10
Output Nodes	2	3	11	2	2	2	3

Figure 2.1. MLP Network Topologies for Benchmark Problems

Topology of LVQ networks for all benchmark data sets were determined as suggested in [31]. The LVQ network had three layers: one input, one competitive and one output layer. The number of hidden layer nodes in the competitive layer was computed using $5 \cdot d$, where d is the dimension of the input patterns. Quantity given by $5 \cdot d^2$ determined the number of weights in the network. The dimensionality of input patterns indicated the number of nodes in the input layer and the number of classes determined the node count in the output layer. Appendix I presents values of other network parameters.

RBF, PNN and PPFNN algorithms have only one heuristically determined parameter, which defines the spread of the radial basis/exponential functions. Values of this parameter for all benchmark problems are presented in Appendix I.

3. Simulation Analysis of PPFNN Performance

A comprehensive comparative simulation analysis of PPFNN with PNN, RBF, MLP and LVQ on the set of benchmark problems has been conducted. The performance criteria included training speed, training and test data classification rate, computational resource requirements, classification rate sensitivity to variations in the value of parameter Alpha (which determines the spread of the exponential functions used in PPFNN and PNN algorithms), incremental learning performance and performance sensitivity to weight generating sequence instance. All neural network algorithms have been simulated using MATLAB Neural Network Toolbox [47]. The harmonic sequence has been employed as the default weight generating sequence.

3.1. Training Time Requirement Analysis

First simulation study involved in measuring the time needed to train neuro-classifiers on the set of benchmark problems. Results of simulation study are presented in Table 3.1.

For two-spiral data set, results in Table 3.1 indicate that the PPFNN algorithm along with RBF and PNN require the minimum amount of training time. The training time of PPFNN algorithm is two minutes, which compares very well with 25 minutes training time for LVQ algorithm, and about 60 minutes training time for MLP algorithm. In the case of IRIS data set, PPFNN and RBF algorithms require the minimum amount of training time. PPFNN algorithm requires 120 seconds to train. This training time requirement is one to two orders of magnitude smaller than the training time requirements of LVQ and MLP. As for the Sonar data set, results presented in Table

3.1 indicate that PPFNN and RBF require the minimum training time. The PPFNN algorithm requires 621 seconds to train, which is close to the performance of the best algorithm, RBF, with a training time of 360 seconds. PNN is a close third while LVQ and MLP perform very poorly. Results for the Vowel data set show that PPFNN and RBF algorithms require the minimum amount of training time. The difference in training times of PPFNN and RBF is only 5%. Table 3.1 indicates that for the Wisconsin Breast Cancer Disease data set, PPFNN algorithm requires the least amount of training time. Training time requirements of PPFNN, RBF and PNN are on the same order. MLP and LVQ neural networks require significantly longer time to train. For the Cleveland Heart Disease data set, PPFNN requires training time on the order of RBF algorithm which offers the best performance. MLP requires the maximum amount of training time. In the case of Thyroid gland disease data set, results in Table 3.1 indicate that PPFNN neural network requires 120 seconds to train: 55 seconds longer that that of the RBF neural network which has the minimum amount of training time.

	2-Spiral	IRIS	Sonar	Vowel	Wisconsin	Cleveland	Thyroid
MLP	3556	12572	47400	52800	10317	8834	9746
LVQ	1500	1320	7800	5237	1653	1835	450
RBF	120	120	360	3600	137	495	65
PNN	120	-	886	-	220	2532	-
PPFNN	120	120	621	3777	98	1010	120

Table 3.1 Training Time (in seconds) Requirements of Neural Network Algorithms on Benchmark Problems.

When all results are considered, neuro-classifiers can be divided into two groups with respect to training time requirements: the first group includes the PPFNN, PNN and RBF and the second group includes MLP and LVQ. Neural algorithms in the second group, MLP and LVQ, perform rather poorly and require lengthy training times. MLP network is well known for its very slow training speed [26][27]. Standard backpropagation learning rule often requires a lengthy training process in which the complete set of training examples is processed hundreds or thousands of times. Neural algorithms in the first group require the minimum training times. Additionally, training time requirement for the PPFNN is on the order of training time requirements of RBF and PNN. The PPFNN has the minimum training time for the IRIS and Wisconsin Breast Cancer data sets. Its training time requirements are second lowest after RBF for Sonar, Vowel, Cleveland Heart Disease and Thyroid Gland Disease data sets: in all cases, the difference in the training time requirements between the RBF and the PPFNN is relatively small. In conclusion, PPFNN training time requirements are either on the order of or less than the training time requirements of all other neuro-classifiers on the set of benchmark problems tested.

3.2. Classification Performance Analysis

In this simulation study, classification rates of five neural classifier algorithms on seven benchmark problems were observed and results are presented in Table 3.2. It is worth noting that the classification performance on test data rather than on the training data is more significant in comparing the performance of different algorithms.

	Training Data Classification Rate in %					Test Data Classification Rate in %				
	MLP	**LVQ**	**RBF**	**PNN**	**PPFNN**	**MLP**	**LVQ**	**RBF**	**PNN**	**PPFNN**
2-Spiral	50.00	55.73	98.96	89.58	91.67					
IRIS	79.00	84.00	100.00	-	98.00	78.00	82.67	80.00	-	96.00
Sonar	59.42	70.51	100.00	89.86	93.35	53.85	62.98	71.15	74.04	73.08
Vowel	46.48	42.53	99.90	-	70.73	36.57	11.11	56.67	-	52.32
Wisconsin	54.97	88.85	100.00	95.33	95.82	59.94	87.88	66.67	95.15	95.76
Cleveland	55.17	61.95	89.66	96.78	99.69	55.17	57.93	65.86	55.86	58.28
Thyroid	36.98	81.98	100.00	-	84.19	36.74	81.86	72.09	-	78.14

Table 3.2 Classification Performances of Neural Network Algorithms on Seven Benchmark Problems.

For Two-Spiral data set, performance of PPFNN algorithm is distinctly superior to the performances of MLP and LVQ neural networks, and its performance is comparable to those of RBF and PNN networks. PPFNN classification rate is on the same order of RBF and PNN while it is significantly better than those of LVQ and MLP algorithms (36% and 42% higher, respectively).

For IRIS training data, classification performances of PPFNN and RBF algorithms are the best. The classification rate of PPFNN is just 2% lower than that of RBF, while it is 19% higher than that of MLP and 14% higher than that of LVQ. Classification performance of PPFNN is the best for IRIS test data set: the classification rate of PPFNN is 16% higher than that of RBF, 18% higher than that of MLP and 13.33% higher than that of LVQ.

For Sonar training data set, classification performance of RBF algorithm is the best. The classification rate of PPFNN algorithm is 6.65% lower than that of RBF, while slightly higher than that of PNN and significantly higher than those of both MLP and LVQ. For the Sonar test data set, classification performance of PPFNN is one of the best and better than that of RBF. Classification rate of PPFNN is comparable to those of PNN and RBF, and significantly better than those of MLP and LVQ.

For the Vowel training data set, performance of PPFNN is the second best after RBF. In the meantime, we can observe that for the Vowel test data set, performance of PPFNN is a close second best after RBF. The difference in classification rate between RBF and PPFNN gets much smaller for the test data compared with the training data. Classification performances of LVQ and MLP worsen specifically for the test data while PPFNN has a good classification performance for test data set in contrast to other algorithms.

The PPFNN algorithm offers the best performance for the test data set while PNN is a close second for the Wisconsin Breast Cancer data set. Although RBF algorithm has the perfect performance on the training data set, its performance quickly degrades to 66.67% for the test data set.

Analysis of the performance of each neural algorithm for the Cleveland heart disease training and test data sets indicates that the classification rate for PPFNN for the test data lags RBF with a percentage difference of 7.58. Classification rates of all algorithms except RBF are within 2 to 3% of within each other.

As for the Thyroid Gland Disease training data set, PPFNN achieves 84.19% classification rate. RBF algorithm has the best classification performance for this particular training data, with a perfect 100% classification rate. Table 3.2 indicates that for Thyroid gland disease test data set, PPFNN algorithm has the second best classification performance, lagging LVQ by 3.72%.

In summary, PPFNN achieves the best classification rates for IRIS and Wisconsin Breast Cancer Disease test data sets. PPFNN scores the second best classification rates for Sonar, Vowel, Cleveland and Thyroid test data sets while closely following the best performer. These results strongly indicate that PPFNN is a top performer when classification rates are considered.

3.3. Computational Resource Requirements

A simulation study to test the validity of the hypothesis which predicts a minimal network structure for the PPFNN algorithm has been conducted. Note that PPFNN, RBF and PNN have identical network structures: on input layer, one hidden layer and one output layer. The number of nodes in the input and output layers for these networks are determined by the dimensionality of the input vectors and number of classes, respectively. However, the number of hidden layer nodes depends on the classification problem instance. Therefore, it becomes meaningful to compare the number of hidden layer nodes for three neural classifiers to assess the computational resource requirements. Simulation results are presented in Table 3.3.

	2-Spiral	IRIS	Sonar	Vowel	Wisconsin	Cleveland	Thyroid
RBF	192	120	192	150	150	201	172
PNN	79	-	68	-	11	137	-
PPFNN	70	10	66	100	5	115	12

Table 3.3 Comparison of the Number of Hidden Layer Nodes on Seven Benchmark Problems.

PPFNN requires the least number of hidden layer nodes for all benchmark problems tested. The computational resource requirements for the PNN is slightly more than what PPFNN requires while RBF consistently utilizes significantly more computational resources. For the cases of IRIS, Wisconsin Breast Cancer and Thyroid Gland Disease data sets, there is a drastic difference in the number of computational resources required by the RBF and PPFNN algorithms. RBF requires noticeably more hidden layer nodes for those tasks. PPFNN captures the clustering properties of the classification data more efficiently: efficiency reflected by the number of hidden layer nodes employed to perform the classification task. This leads to PPFNN requiring the minimal network structure compared to RBF and PNN.

3.4. Sensitivity Analysis of PPFNN Classification Rate with Respect to Parameter Alpha

Parameter Alpha determines the spread of potential functions placed around a training pattern in the learning phase. Therefore, it is important to study the effect of this parameter on PPFNN performance. In the following simulation experiments, a range of values for the parameter Alpha has been tested to optimize the classification performance of PPFNN and PNN algorithms.

Simulation results for the two-spiral data set are presented in Figure 3.1. Results demonstrate that PPFNN and PNN algorithms are not highly sensitive to the variations in the value of Alpha. For large Alpha variations (from 0.1 to 10), the classification rate varies only a small amount (from 90.62% to 91.67% for PPFNN, from 87.50% to 89.58% for PNN). The two curves in Figure 3.1 are almost flat and PPFNN consistently has a better performance than that of PNN at each value of Alpha.

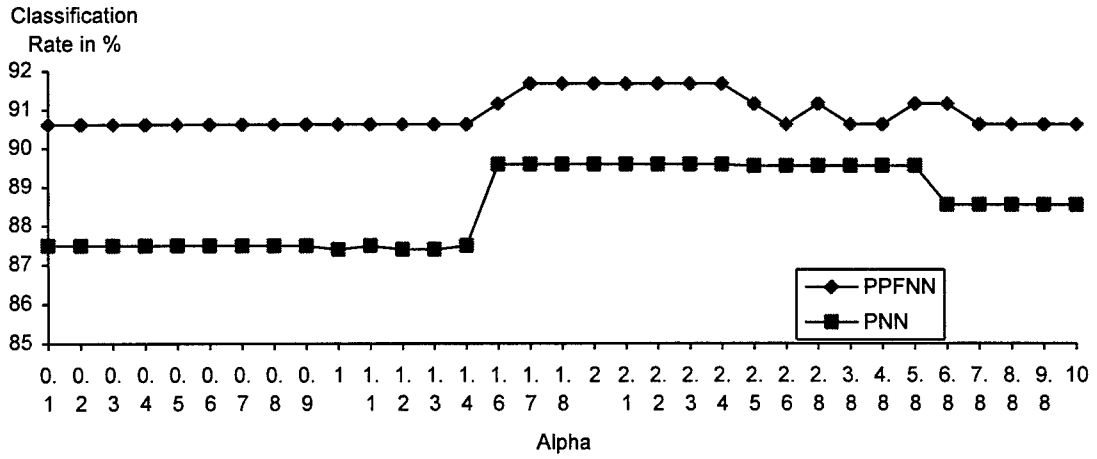


Figure 3.1. Classification Rates of PPFNN and PNN on Two-Spiral Data Set.

In the case of Sonar test data set, Figure 3.2, PPFNN and PNN classification rates are not highly sensitive to the variations in values of Alpha. As Alpha varies in the range 2.5 to 15, the classification rate varies only a small amount. For the test data set, classification rate of PPFNN varies from 69.23% to 73.08%, and that of PNN varies from 68.75% from 74.04%. A nonuniform scale was used to highlight the Alpha values from 4.2 to 4.5. The classification rate curve for both neural algorithms oscillate in the range tested. PPFNN offers relatively higher classification rates for Alpha values at or around 4.4 and 12.5.

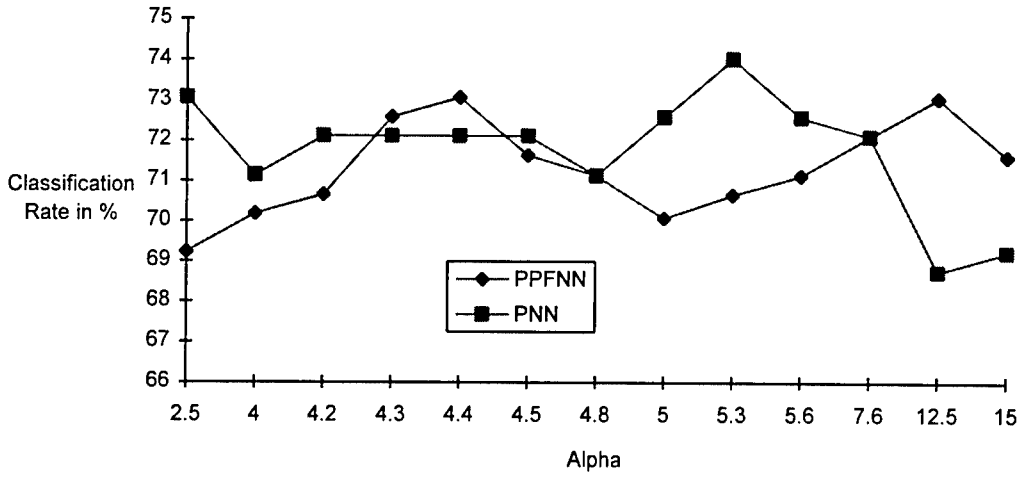


Figure 3.2. Classification Rates of PPFNN and PNN on Sonar Test Data Set.

Simulation results for the Vowel test data set, Figure 3.3, indicate that PPFNN algorithm is not highly sensitive to large variations of Alpha values and the two curves are almost flat as Alpha varies in the range 8.5 to 9.3 (a non-uniform scale is used to emphasize the Alpha values from 8.5 to 9.3). For the entire Alpha range (3 to 15), the maximum change in the classification rate is 3.33%: classification rate varies from 48.99% to 52.32%. It is also worthwhile to note that larger values of Alpha results in higher classification rates.

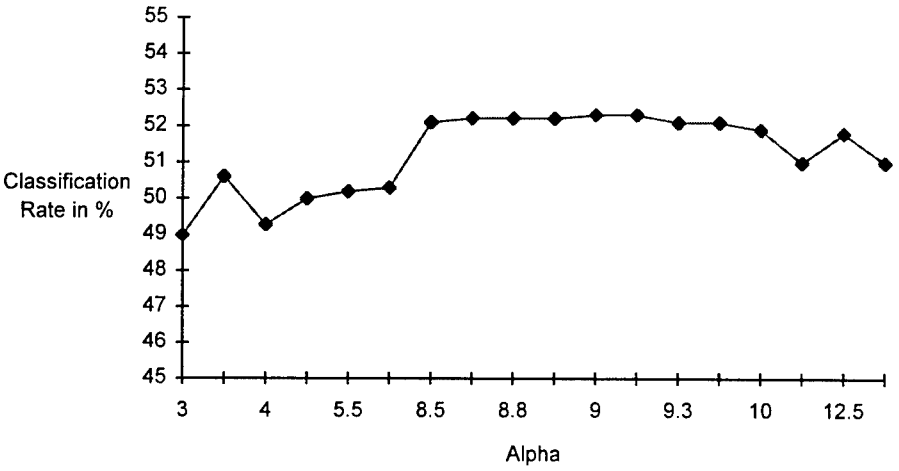


Figure 3.3. Classification Rates of PPFNN on Vowel Test Data Set.

Results for Wisconsin breast cancer test data set are shown in Figure 3.4. Classification rates of PNN and PPFNN change a maximum of 5% in response to variations in Alpha values from 0.8 to 12.8. Classification rate is consistently higher for small values of Alpha for both algorithms. It is also interesting to note that PPFNN and PNN curves are correlated to a large degree in the figure.

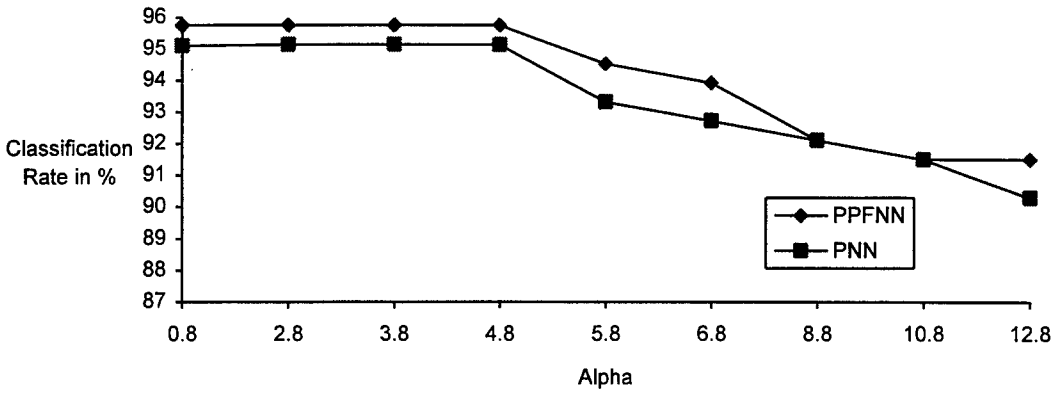


Figure 3.4. Classification Rates of PNN and PPFNN on Wisconsin Breast Cancer Disease Test Data Set.

Figure 3.5 shows the simulation results on Cleveland Heart Disease test data set. These results are for Alpha values in the range from 1.8 to 18.8. The PPFNN test data classification rate changes from 44.48% to 58.28%: a relatively large variation. For the PNN algorithm, the test data classification rate changes from 45.86% to 55.85%. In overall, PPFNN classification rate varies as much as 14% for the values of Alpha in the testing range indicating that PPFNN performance is sensitive to values of Alpha for this data set. Larger values Alpha results in better classification rate values for PPFNN. On the other hand, the PNN curve indicates that the classification rate is higher for small values of Alpha.

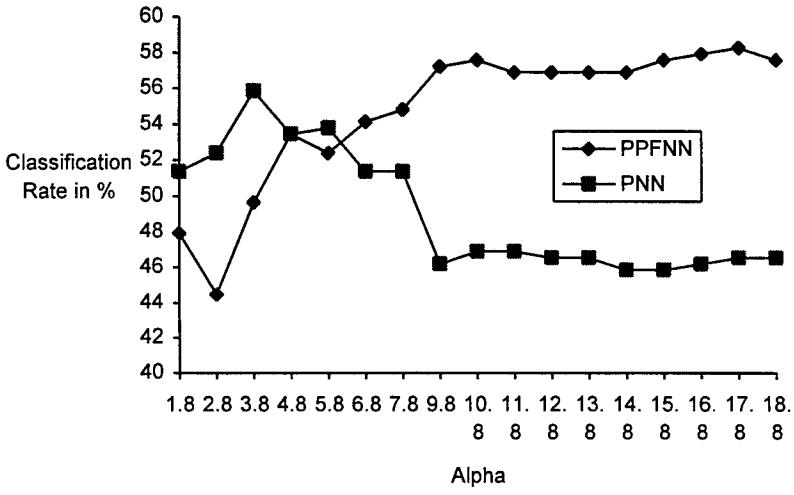


Figure 3.5. Classification Rates of PPFNN and PNN on Cleveland Heart Disease Test Data Set.

For Thyroid Gland Disease test data set, simulations were conducted with Alpha values in the range from 0.0012 to 0.015, Figure 3.6. Results demonstrate that PPFNN algorithm classification rate is highly sensitive to

changes in Alpha. When Alpha value varies from 0.0012 to 0.015, the test classification rate changes from 50.23% to 78.14%. The change for this benchmark problem is significantly large, which is about 28% and larger values of Alpha provides higher classification rates for PPFNN.

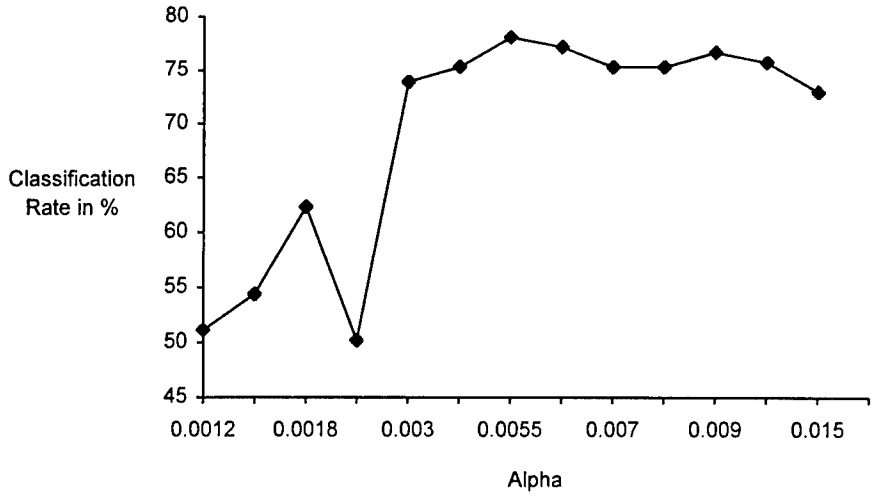


Figure 3.6. Classification Rates of PPFNN on Thyroid Gland Disease Test Data Set.

In conclusion, PPFNN classification rates were not found to be highly sensitive to variations in the value of Alpha for Two-Spiral, Sonar, Vowel and Wisconsin Breast Cancer data sets: maximum change observed in classification rate values were in the range of 5%. Classification rates changed significantly in response to changes in the value of Alpha for Cleveland Heart Disease and Thyroid Gland Disease data sets: a maximum of 14% and 28% variation, respectively. Results indicate that Alpha plays an important role for the classification performance of PPFNN and an initial search to determine a "good" value for this parameter is needed.

3.5. Incremental Learning Performance

In this simulation study, classification performance of neural classifiers were observed against growing training data set sizes. The MLP neural network was excluded from this study since its training time requirements are excessively large. Initially, the overall data set for a given benchmark problem was divided into two sections: training data and testing data. Furthermore, the training data was evenly divided into multiple groups. First, the neuro-classifiers were all trained on a single group and their classification performances were measured on the test data set. Next, another group of training data was added to the training set and classification performances of neural classifiers were observed. This process was repeated until all groups in the training data set were included in the training process. The set of benchmark problems employed in this simulation study are Sonar, Vowel, Wisconsin

Breast Cancer, and Thyroid Gland Disease data sets. PPFNN, PNN, RBF, and LVQ have been employed as neural network classifiers.

The Sonar data set was divided into 13 groups: twelve groups for training and one group for testing. Initially only one group of Sonar data, out of twelve available, was used for training and the thirteenth group was used for testing. Results in Figure 3.7 indicate that PPFNN achieves a comparable classification rate with respect to other neural classifiers starting with the initial training and throughout the overall experiment. Classification performances of PNN, RBF and LVQ all fluctuate as new training data is added while the performance of PPFNN consistently improves: each new addition of training data contributes to the classification performance.

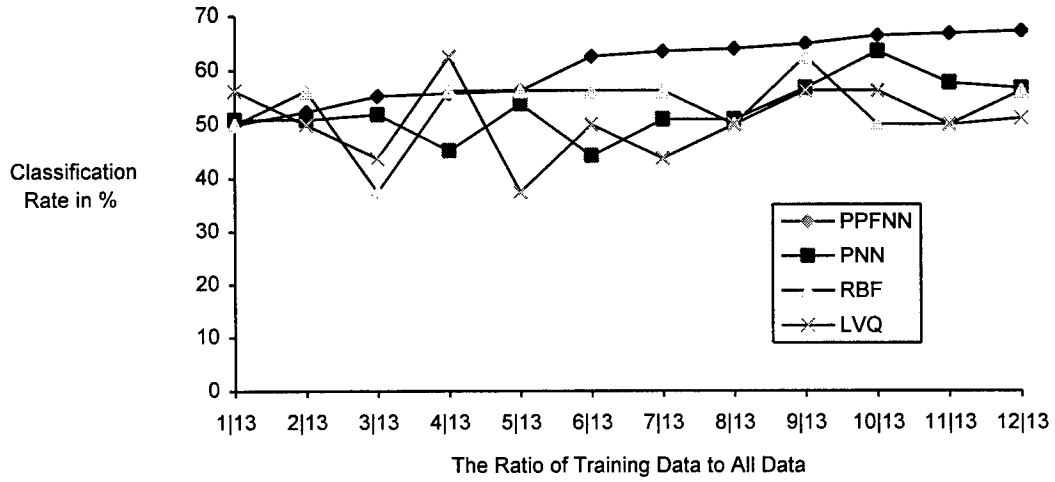


Figure 3.7. Incremental Learning Performances of PPFNN, PNN RBF and LVQ on Sonar Data.

Next simulation test case involved assessing the incremental learning performance of PPFNN, RBF and LVQ on Vowel data set, which was evenly divided into 6 groups. Five groups were used for training and the sixth group was used for testing. Simulation results are presented in Figure 3.8. PPFNN captures the clustering properties of benchmark data better than the other two algorithms for small training set sizes. PPFNN initially starts at 40% before it reaches approximately 53.21% classification rate for full training data set.

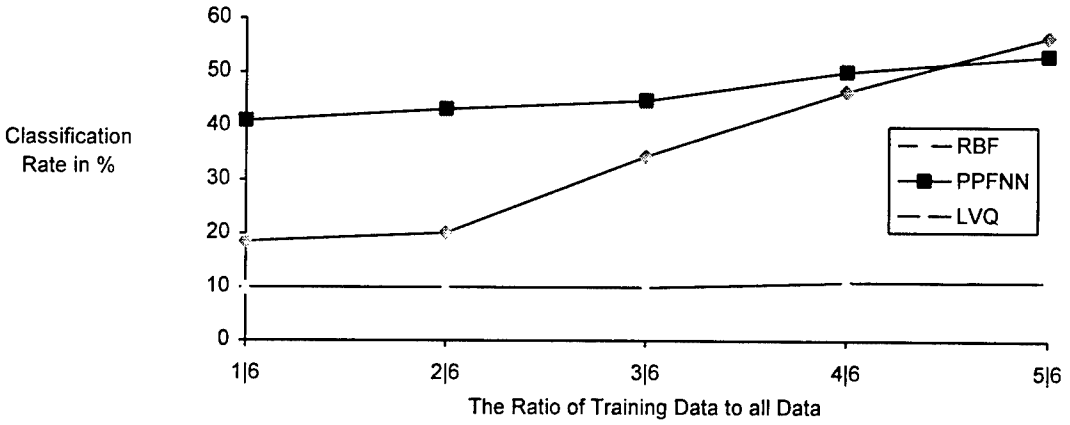


Figure 3.8. Incremental Learning Performances of RBF, PPFNN and LVQ on Vowel Data.

Classification performances of neuro-classifiers including PPFNN on Wisconsin Breast Cancer data set were observed in the next simulation experiment. Simulation results are presented in Figure 3.9. The Breast Cancer data set was divided into 10 groups. The tenth group was used as testing data while the remaining nine groups were used as training data. Simulation results in Figure 3.9 indicate that, PPFNN along with LVQ and PNN, were able to capture the clustering properties of the data set with training only on one-tenth of the overall data set. However, LVQ and PNN do a distinctly superior job in learning the properties of the classification problem with minimum number of training patterns. Classification rates did not vary appreciably as more data was added to training data set for those algorithms.

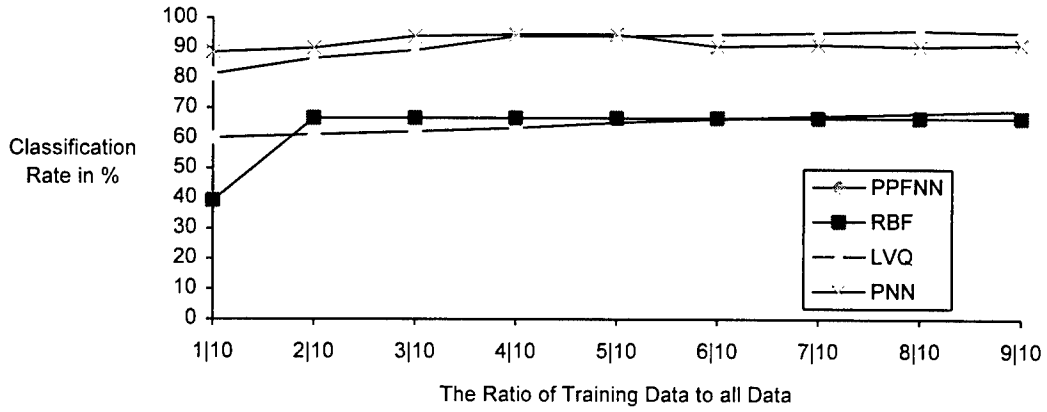


Figure 3.9. Incremental Learning Performances of PPFNN, PNN, RBF and LVQ on Wisconsin Breast Cancer Disease Data.

Figure 3.10 shows the classification performances of PPFNN, LVQ and RBF on Thyroid Gland Disease data set. The Thyroid Gland Disease data set was evenly divided into 5 groups: four groups for training data set and

one group as the test data. Figure 3.10 indicates that LVQ algorithm can learn the clustering properties of Thyroid Gland Disease data set for small training set sizes. When more training data was added, the classification rate on test data improved for LVQ and PPFNN algorithms. On the other hand, for RBF algorithm, when more data was added to the training data, the classification performance on test data became worse.

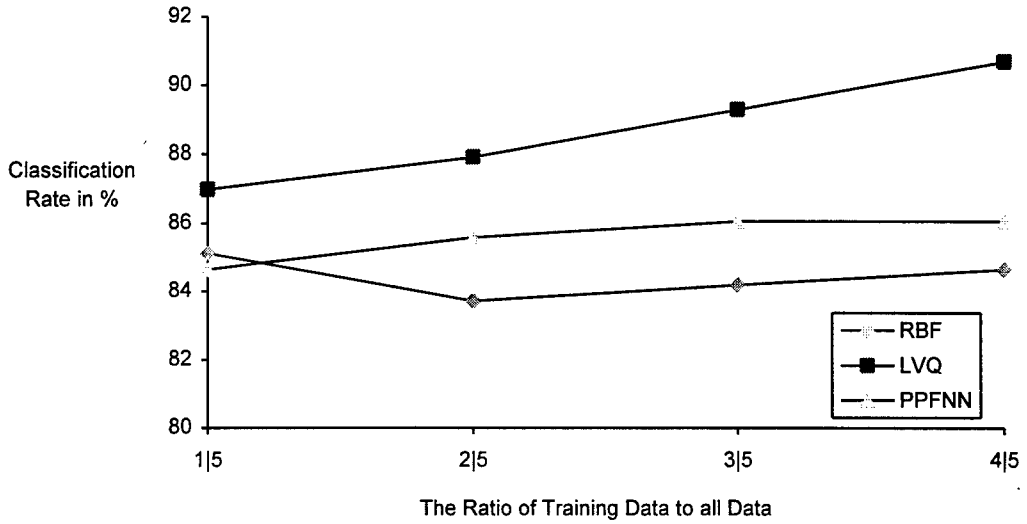


Figure 3.10. Incremental Learning Performances of PPFNN, LVQ and RBF on Thyroid Gland Disease Data.

In conclusion, PPFNN achieved to discover the clustering properties of the benchmark data sets with a training on a fraction of the complete data set. PPFNN steadily improved its classification rate as more data was included in the training set although the improvements have not been drastic for Sonar, Vowel and Wisconsin Breast Cancer Disease data sets. The classification rate for PPFNN dropped approximately 1% for increased training data sets. Variations in the classification rates for PNN, RBF and LVQ are larger when all four problems are considered.

3.6 Effect of Weight Sequence on PPFNN Performance

Mathematical description of PPFNN requires the weights to be generated by a sequence which satisfies the following three conditions:

$$1)\ \lim_{k\to\infty}\gamma_k = 0,\quad 2)\ \sum_{k=1}^{\infty}\gamma_k = \infty,\quad 3)\ \sum_{k=1}^{\infty}\gamma_k^2 < \infty \tag{3.1}$$

A large set of sequences will satisfy these conditions and therefore, qualify to become a weight generating sequence for the PPFNN algorithm. The choice of the weight generating sequence is likely to affect the performance of PPFNN. The harmonic sequence has been utilized in all of the simulation work up to this point with satisfactory performance results for PPFNN. However, it is possible to choose a different sequence which may improve the

overall performance of PPFNN. In order to test this hypothesis, four distinct weight generating sequences will be employed to observe the performance of PPFNN on a set of benchmark problems. Sequences used are defined by:

Harmonic Sequence	$\gamma_k = \frac{1}{k}$
Sequence 1	$\gamma_k = \frac{1}{k^{0.75}}$
Sequence 2	$\gamma_k = \frac{(\ln k)^{1.5}}{k}$
Sequence 3	$\gamma_k = \frac{(\ln k)^{2.5}}{k}$

where $k = 1, 2, 3, \ldots$

These sequences have been specified with respect to their convergence speed to zero. Harmonic sequence is the fastest converging sequence followed by sequence 1, sequence 2, and sequence 3. Three measures of performance have been observed for all simulation experiments: classification rate, training time and the number of nodes in the hidden layer. The set of benchmark classification problems included Sonar, Vowel, Wisconsin Breast Cancer, Cleveland Heart Disease and Thyroid Gland Disease data sets. Classification rates for test data for all benchmark problems included in this study are shown in Figures 3.11 through 3.15.

For Sonar data set, difference in classification rates for the set of weight generating sequences is limited to a maximum of approximately 5% as shown in Figure 3.11. Data in Figure 3.11 indicates that the sequence 3 leads PPFNN steadily to better performance values in a large interval of Alpha. However, performance plots for all weight generating sequences are clustered throughout the Alpha testing interval.

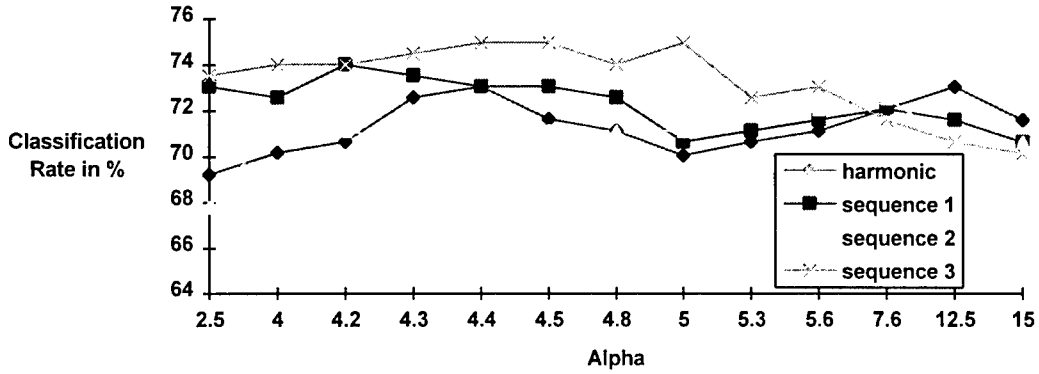


Figure 3.11. Performance of PPFNN on Sonar Data vs. Weight Generating Sequences.

Results for Vowel data are mixed, Figure 3.12. Harmonic sequence is the top performer for the region where Alpha values are approximately between 8.5 and 15. For smaller Alpha values, performance curves oscillate

without declaring a clear winner. In the overall Alpha range, [3, 15], the maximum change in the PPFNN performance due to weight generating sequence is not more than 5%.

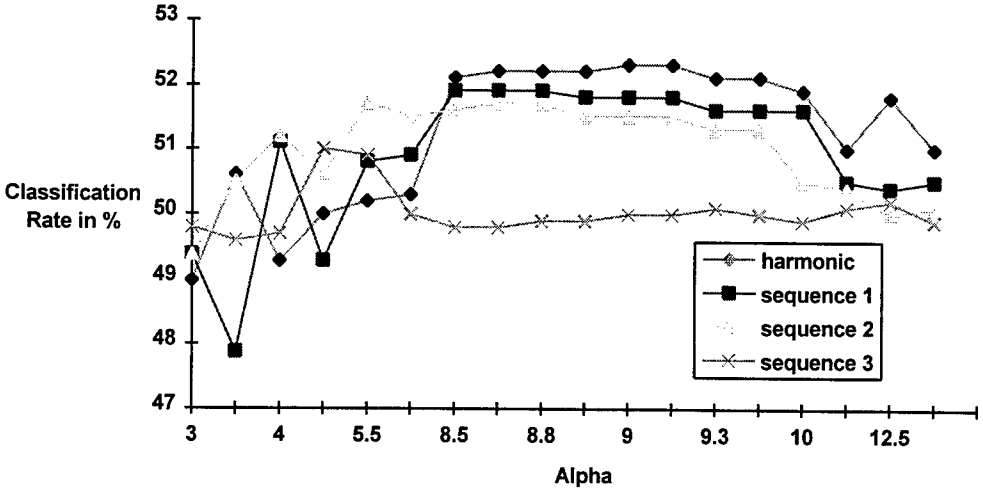


Figure 3.12. Performance of PPFNN on Vowel Data vs. Weight Generating Sequences.

Performance curves in Figure 3.13 indicate that PPFNN performance is not affected by the choice of weight generating sequence for Wisconsin Breast Cancer data set. There are only minor differences in performance values typically limited to no more than 1%. It is worth noting that all four performance curves drop approximately 8% for larger values of Alpha.

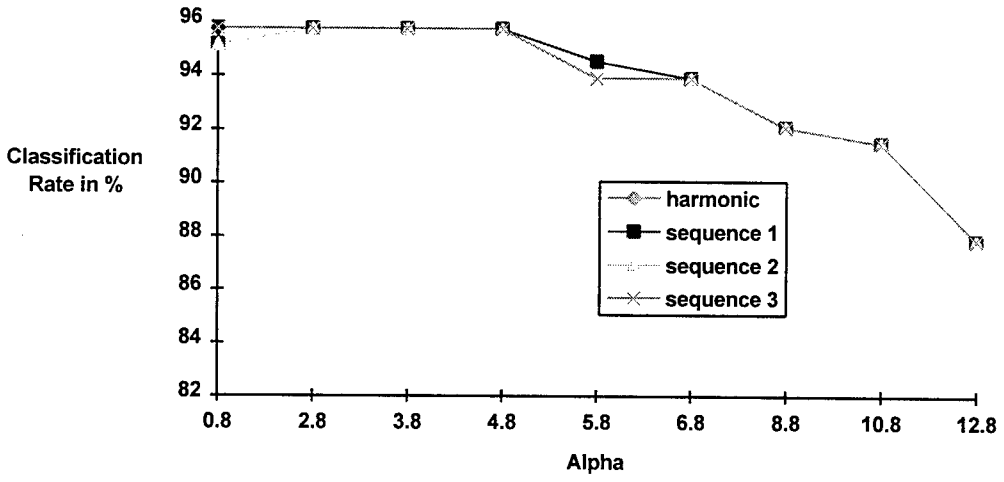


Figure 3.13. Performance of PPFNN on Wisconsin Breast Cancer Data vs. Weight Generating Sequences.

PPFNN classification rates for all four weight generating sequences overlap throughout the range of Alpha values tested for Cleveland Heart Disease data set, Figure 3.14. Simulation results indicate that performance of PPFNN is independent of the choice of weight generating sequence for Cleveland Heart Disease data set.

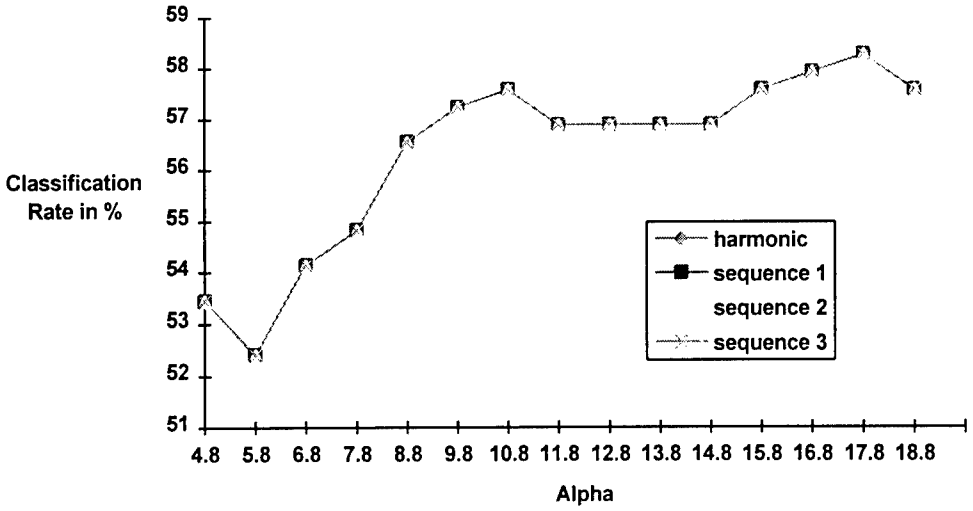


Figure 3.14. Performance of PPFNN on Cleveland Heart Disease Data vs. Weight Generating Sequences.

Performance of PPFNN on Thyroid Gland Disease data set with respect to the weight generating sequences is presented in Figure 3.15. Performance curves converge to each other for Alpha values greater than approximately 0.3 and stay together for larger values of Alpha. However, the difference in performance curves reaches 50% in the interval where Alpha is between 0.001 and 0.3. Although performance curves due to harmonic sequence, sequence 1 and sequence 2 are clustered, the performance curve due to sequence 3 drops significantly to 40% classification rate marker for very small values of Alpha. It is important to note that performance curves due to harmonic sequence, sequence 1 and sequence 2 all climb to higher values of classification rates as Alpha values decrease.

In conclusion, the effect of weight generating sequence instance on the classification rate of PPFNN has translated into a no more than 5% performance variation for all benchmark problems employed with the exception of Thyroid Gland Disease data set. Classification rate of PPFNN dropped approximately 50% for Thyroid Gland Disease data set when sequence 3 has been employed. In general, PPFNN performance has been consistently better with the harmonic sequence and sequence 1 and the performance curve due to sequence 3 has lagged other performance curves for almost all of the benchmark problems.

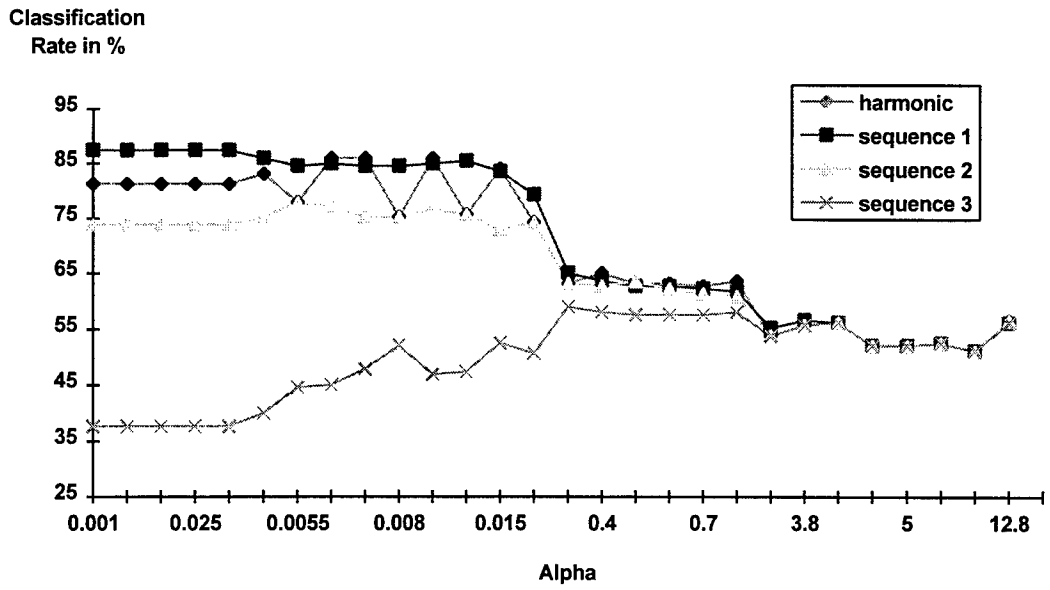


Figure 3.15. Performance of PPFNN on Thyroid Gland Disease Data vs. Weight Generating Sequences.

Effect of weight generating sequence on the training time requirements of PPFNN algorithm for a select set of benchmark problems is presented in Table 3.4. In the case of Sonar, Vowel, Wisconsin Breast Cancer and Cleveland Heart Disease data sets, the training time requirements vary by as much as 50%. The change in training time requirements is more drastic in the case of Thyroid Gland Disease data set: the training time reduces by more than 90%. Sequence 3 consistently offers the best training times while the training time requirements are the largest for the Harmonic sequence. However, sequences which require PPFNN to spend less time in training are those which offer poor classification performances in general.

Sequences	*Sonar*	*Vowel*	*Wisconsin*	*Cleveland*	*Thyroid*
Harmonic	621	3777	98	1010	215
Sequence 1	581	2900	90	987	42
Sequence 2	406	2000	60	670	37
Sequence 3	330	1600	42	450	14

Table 3.4. Training Time Requirements (in Seconds) of PPFNN on Benchmark Problems for the Set of Weight Generating Sequences.

Computational resources needed, specifically the number of hidden layer nodes, is also affected by the choice of the weight generating sequences for Vowel, Cleveland Heart Disease and Thyroid Gland Disease data sets, Table 3.5. The reduction in number of hidden layer nodes is almost 10% for Vowel and Cleveland Heart Disease data sets and 85% for Thyroid Gland Disease data set. The change in computational resources is insignificant for the remaining data sets.

	Hidden Layer Node Count				
Sequences	*Sonar*	*Vowel*	*Wisconsin*	*Cleveland*	*Thyroid*
Harmonic	66	104	6	115	40
Sequence 1	65	100	5	109	24
Sequence 2	63	90	4	102	14
Sequence 3	63	90	4	100	5

Table 3.5. Computational Resource Requirements of PPFNN on Benchmark Problems for the Set of Weight Generating Sequences.

In conclusion, choice of weight generating sequence affects the classification rates, the training time and the number of hidden layer nodes of PPFNN for the set of problems studied. However, those weight generating sequences which lead PPFNN to better classification rates require more training time and computational resources. Harmonic sequence and Sequence 1 offer the best classification rates while Sequence 3 requires the least amount of training time and the minimum number of hidden layer nodes.

3.7 Summary

Simulation results presented in this section demonstrate that PPFNN performance is comparable to and often better than those of MLP, RBF, LVQ and PNN algorithms when the set of seven benchmark problems are considered. PPFNN scored minimal training time requirements for the set of problems tested: PPFNN has a fast learning speed and can be trained on-line. Classification rates PPFNN achieved on the set of benchmark problems equaled or surpassed the top performing neuro-classifiers including RBF and PNN. The number of hidden layer nodes required by PPFNN was minimal compared to the rest of the neuro-classifiers tested: this empirical result supports the theoretical prediction that PPFNN is likely to adapt to a minimal network topology. In most cases, PPFNN showed a very good ability to capture the clustering properties of the classification problem with minimal training data and improved its performance as the size of the training data set grew. This observation opens the way for successfully employing PPFNN with NREDS data set since the size of the initial training data set is typically small for that task. PPFNN performance has been robust with respect to changes in the weight generating sequence instance for most of the data sets tested. Harmonic sequence consistently scored well for the full range of benchmark problems. It is possibly a good choice for the weight generating sequence as the starting point. The value of Alpha, a heuristically determined parameter, was not a significant factor affecting the performance of PPFNN for most of the problems studied. PPFNN algorithm showed a robust performance with respect to two empirically determined parameters, the weight generating sequence and Alpha.

4. Simulation-Based Performance Evaluation of PPFNN on Actual NREDS Data

4.1 Introduction

This section is devoted to studying the classification performance of PPFNN on actual NREDS data set [5][6][7][8]. NREDS operational environment is characterized by a lack of sufficiently large training data set. Therefore, the classification algorithm needs to be able to capture clustering properties of the problem using the limited training data. In this section, PPFNN classification performance on limited NREDS data set will be observed. Actual NREDS data used in the simulation study is presented in Appendix 3 of reference [51].

4.2 NREDS Data Description

The data file coming from the flight data recorder box contains groups of records, each group corresponding to an individual circuit card, Figure 4.1. Because there are multiple image sequences of the same card in varying states of disrepair, the distinction between these image sets is made by a time and date stamp, which is placed on the first record of the record group. The first record of the record group begins with a control character (ASCII 127, the home plate symbol). After this symbol comes the date and time stamp, " 960421.001327," which states the image was collected on the 21st of April in 1996, at 00:13:27. Immediately after the time and date stamp comes the card serial number, 1447X1.

```
Part ID
r  left   top  right bottom heat
960421.001327 1447X1
CLASS: *U3
1  8136  1258 14866  2096   .0013 DA U2        .0000   .0034
1 10830  2794 15076  3576   .0358 EA U3        .0000   .0029
1  8224  4340 15020  5148   .0000 FA U1        .0000   .0040
```

Figure 4.1. Part of NREDS Data Set.

The classification, or outcome of the circuit card, if known, is placed on the record immediately following the time/date stamp, in particular, CLASS: *U3. The subsequent records contain the heat rates and alignment information corresponding to the individual component on the circuit card, one record for each component. The heat rates are the inputs to the neural network. This may be confusing because the record also contains component identification and alignment information and additional computed information about the expected tolerance range for the heat rate. The heat rate is the sixth data field on the record. The first five data fields deal with the component alignment; they are not germane to the neural process. The next two fields are component label designators. The final two entries are the heat rate tolerance range which is derived from two good cards.

4.3 Simulation Results

PPFNN can implement incremental learning procedure and hence, does not disturb the previous state of the network but simply add new computational resources to the existing network topology to learn the new training pattern. This property results in generally smaller training time requirements compared to other neuro-classifiers including MLP, LVQ, PNN and RBF as demonstrated by simulation results on benchmark problems. Same simulation experiments on benchmark problems also indicated that PPFNN is comparably better in capturing the clustering properties of a given problem even when training data is limited to a fraction of the actual data set: see Figures 3.7 through 3.10. In the simulation analysis to follow, the classification performance of PPFNN, as more and more NREDS training data becomes available, will be observed.

The NREDS data set had a total of 15 patterns. These 15 patterns were divided into three groups evenly and randomly. Let these three groups be called Group 1, Group 2 and Group 3. The performance testing procedure implemented is given as follows:

- Designate one group out of three as the test data set.
- Designate the remaining two groups as training data sets.
- Use one group out of two training groups to train the network.
- Test classification performance using the test data set.
- Use the second training group for additional training.
- Test the classification performance again using the test data set.

Three test cases were implemented as follows:

Case 1: Training set {Group 1, Group 2}, Test set {Group 3}
Case 2: Training set {Group 1, Group 3}, Test set {Group 2}
Case 3: Training set {Group 2, Group 3}, Test set {Group 1}

Simulation results are presented in Table 4.2.

	PPFNN Classification Rate Training Data Set Size	
	5 patterns	**10 patterns**
Case 1	40.00%	80.00%
Case 2	20.00%	60.00%
Case 3	40.00%	60.00%
Average	33.33%	66.67%

Table 4.2. The Incremental Learning Performance of PPFNN on NREDS Data Set.

Results in Table 4.2 indicate that PPFNN captures the clustering properties of NREDS data with a minimal size training data set. Similar experiments run on benchmark data sets also provided results which supported this conclusion. It is expected that PPFNN will show a performance similar to its performances on benchmark data sets for the NREDS data set as more comprehensive data becomes available. Field test results of PPFNN are also very encouraging as reported by Allred [50].

5. Conclusions

This study was devoted to simulation-based investigation of the performance of a newly proposed neural network algorithm, PPFNN. A comprehensive simulation analysis of PPFNN algorithm on a set of benchmark problems has been performed. Benchmark problems tested include IRIS data set, Sonar data set, Two-Spiral data set, Vowel data set, Wisconsin Breast Cancer data set, Cleveland Heart Disease data set and Thyroid Gland Disease data set. Performance of PPFNN has been compared to those of MLP, LVQ, RBF and PNN on these benchmark problems. Simulation studies involved measuring the training speed, training data classification rate, test data classification rate, computational resource requirements, effect of heuristically determined parameter Alpha, effects of weight generating sequence instances on the classification performance and dependence of classification rates on training set size. Additionally, a similar study was performed on NREDS data.

The training time requirement for PPFNN is on the order of training time requirements of PNN and RBF in general. The PPFNN offers the minimum training time for the IRIS and Wisconsin Breast Cancer data sets. Its training time requirements are second lowest after RBF for Sonar, Vowel, Cleveland Heart Disease and Thyroid Gland Disease data sets. In all cases, the difference in the training time requirements between RBF and PPFNN is relatively small. In overall, simulation results demonstrate that PPFNN requires minimal training time when a family of neural classification algorithms including MLP, RBF, LVQ and PNN are considered.

Classification performance of PPFNN tops all other algorithms for IRIS and Wisconsin Breast Cancer data sets. It is a close second to the top performing neuro-classifier algorithm for Sonar, Vowel, Cleveland Heart Disease and Thyroid Gland Disease data sets. In all, classification performance of PPFNN is definitely superior to that of MLP and LVQ and comparable to that of RBF and PNN.

The computational resource requirements for PPFNN is the least for IRIS, Sonar, Wisconsin Breast Cancer and Thyroid Gland Disease data sets. It is also important to note that the number of hidden layer nodes required by PPFNN is typically half of what the second best neural classification algorithm requires. PPFNN is second to RBF and PNN for Vowel and Cleveland Heart Disease data sets, respectively, in terms of computational resources needed.

PPFNN performance was not affected by large variations in the value of the heuristically determined parameter Alpha which determines the spread of the potential functions. The only exception was the case of Thyroid Gland Disease data set where the test classification rate varied as much as 27.91% and the training classification rate varied as much as 31.40% as Alpha was changed in the interval [0.0012, 0.015].

In an another simulation study, the classification rate versus the size of the training data set was observed: specifically, the issue of interest was the ability of PPFNN to capture the clustering properties of the classification problems with minimal training data. Results clearly indicate that PPFNN captured the clustering properties of the classification problems after it was trained only with anywhere from one-fifth to one-tenth of the training patterns in the original training data set.

Simulation experiments for the effect of the weight generating sequences on the performance of PPFNN do not indicate that there is a preferable weight generating sequence which will always result in better performance measure values across the set of benchmark problems tested. A set of four distinct weight generating sequences were employed to measure their effect on PPFNN classification performance. Sequence 1 has consistently performed among the top two sequences which also included the harmonic sequence. Simulation results are inconclusive to judge the validity of the hypothesis that the convergence speed of the weight generating sequence is an important factor for the classification rate of PPFNN. The effect of the weight generating sequences on the training time requirements of PPFNN indicates that sequences which require PPFNN to spend less time in training are those which lead to poor classification performance in general. A similar conclusion can be reached for the effect of the weight generating sequences on the computational resource requirements of PPFNN: sequences 2 and 3 result in a drastic reduction in the number of hidden layer nodes needed although those sequences lead to poor performance values for the classification rate.

Simulation results indicate that PPFNN either led other neuro-classifier algorithms or closely followed them in terms of the set of performance measures employed for the set of benchmark problems. This leads to the conclusion that PPFNN is a suitable choice for stochastic pattern classification problems.

Bibliography

[1] Werbos, P.J., "Links Between Artificial Neural Networks (ANN) and Statistical Pattern Recognition" in Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections, I.K.Sethi and A.K. Jain (Editors), pp. 11-31, 1991.

[2] Pao, Y-H., Adaptive Pattern Recognition and Neural Networks, Addison-Wesley Publishing Company, reading: MA, 1989.

[3] J. T. Tou and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley Publishing Company, Advanced Book Program / World Science Division.

[4] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice Hall, 1978.

[5] Lloyd G. Allred and Thomas R. Howard, Application of Thermal Imaging to Electronic Fault Diagnosis, SPIE Conference Proceedings, Orlando, 1994.

[6] Lloyd G. Allred and Gary E. Kelly, A Modified Genetic Algorithm for Extracting Thermal Profiles from Infrared Image Data, SPIE Conference Proceedings, 1992.

[7] Lloyd G. Allred and John D. Gilchrist, Diagnosis of Faults in Electronic Circuits by Integrating Neural Network technology with Thermal Imaging, Symposium on Artificial Intelligence Applications, 1992.

[8] Lloyd G. Allred, Thomas R. Howard and Gursel Serpen, On the Fly Neural Network Construction for Repairing F-16 Flight Control Panel Using Thermal Imaging, SPIE, Vol. 2766, pp. 284-294, 1996.

[9] Scott E. Fahlman and Christian Lebiere, The Cascade-Correlation Learning Architecture, Advances in Neural Information Processing Systems, Vol. 2, pp. 524-532, 1990.

[10] Douglas L. Reilly, Leon N. Cooper and Charles Elbaum, A Neural Model for Category Learning, Biological Cybernetics, Vol. 45, pp. 35-41, 1982.

[11] Michael R. Berthold and Jay Diamond, Boosting the Performance of RBF Networks with Dynamic Decay Adjustment, Advances in Neural Information Processing, Cambridge MA, 1995, 7, pp. 521-528, the MIT Press.

[12] Michael R. Berthold, The TDRBF: A Shift Invariant Radial Basis Function Network, Proceedings of the Irish Neural Network Conference, pp. 7-12, 1994.

[13] Michael R. Berthold, A Probabilistic Extension for the DDA Algorithm, available at http://www.fzi.de/divisions/acid/staff/berthold/papers.html.

[14] Michael J. Hudak, RCE Classifier: Theory and Practice, Cybernetics and Systems, Vol. 23, pp. 483-515, 1992.

[15] John B. Hampshire and Alexander H. Waibel, A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Neural Networks, Vol. 1, No. 2, pp. 216-228, June 1990.

[16] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano and Kevin J. Lang, Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No. 3, pp. 328-339, March, 1989.

[17] Teuvo Kohonen, Improved Versions of Learning Vector Quantization, IJCNN Proceedings, Vol. 1, pp. 545-550, 1991.

[18] Nikhil R. Pal, James C. Bezdek and Eric C.-K. Tsao, Generalized Clustering Networks and Kohonen's Self-Organizing Scheme, IEEE Transactions on Neural Networks, Vol. 4, pp. 549-556, No. 4, July 1993.

[19] John Moody and Christian J. Darken, Fast Learning in Networks of Locally-Tuned Processing Units, Neural Computation, Vol. 1, pp. 281-294, 1989.

[20] Yi Shang and Benjamin W. Wah, Global Optimization for Neural Network Training, Computer, pp. 45-54, March 1996.

[21] Donald F. Specht, Probabilistic Neural Networks, Neural Networks, Vol. 3, pp. 109-118, 1990.

[22] Roy L. Streit and Tod E. Luginbuhl, Maximum Likelihood Training of Probabilistic Neural Networks, IEEE Transactions on Neural Networks, Vol. 5, No. 5, pp. 764-7783, September, 1994.

[23] Sukhan Lee and Rhee M. Kil, A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning, Neural Networks, Vol. 4, pp. 207-224, 1991.

[24] Donald F. Specht, Probabilistic Neural Networks and the Polynomial ADALINE as Complementary Techniques for Classification, IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 111-121, March, 1990.

[25] Emanuel Parzen, On Estimation of A Probability Density Function and Mode, pp. 1065-1076, Standford University, 1972.

[26] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus and Shimon Schocken, Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function, Neural Networks, Vol. 6, pp. 861-867, 1993.

[27] Martin Riedmiller and Heinrich Braun, A Direct Adaptive Method for Faster Backpropagation learning: The RPROP Algorithm, IEEE International Conference on Neural Networks, pp. 586-591, 1993.

[28] Dimitris A. Karras and Stavros J. Perantonis, An Efficient Constrained Training Algorithm for Feedforward Networks, IEEE Transactions on Neural Networks, Vol. 6, No. 6, pp. 1420-1433, November 1995.

[29] Luigi Pietro Cordella, Claudio De Stefano, Francesco Tortorella, and Mario Vento, A Method for Improving Classification Reliability of Multilayer Perceptrons, IEEE Transactions on Neural networks, Vol. 6, No. 5, pp. 1140-1147, September 1995.

[30] Scott E. Fahlman, Fast-Learning Variations on Back-Propagation: An Empirical Study, pp. 38-51, Proceedings of the 1988 Connectionist Models Summer School.

[31] Donald F. Specht and Philip D. Shapiro, Generalization Accuracy of Probabilistic Neural Networks Compared with Back-Propagation Networks, ICNN Proceedings, Vol. 1, pp. 887-892, 1991.

[32] Ning Qian and Terrence J. Sejnowski, Predicting the Secondary Structure of Globular Proteins Using Neural Network Models, J. Mol. Biol., Vol. 202, pp. 865-884, 1988.

[33] Werbos, P.J., The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, John Wiley & Sons, Inc. New York, NY, 1994.

[34] Gursel Serpen, An Artificial Neural Network Classifier for Multi-Modal Distributed Classes, AFOSR SFRP Report, pp. 14.1-14.20, November 1995.

[35] Gursel Serpen, Lloyd G. Allred and Krzysztof J. Cios, Probabilistic Potential Function Neural Network Classifier, ICNN, 96 Proceedings, Vol. Special Sessions, pp. 193-198, 1996.

[36] E. M. Braverman, On the Method of Potential Functions, Avtomatika i Telemekhanika, Vol. 26, No. 12, pp. 2205-2213, December, 1965.

[37] M. A. Aizerman, E. M. Braverman and L. I. Rozonoer, Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning, Avtomatika i Telemekhanika, Vol. 25, No. 6, pp. 917-936, June, 1964.

[38] M. A. Aizerman, E. M. Braverman and L. I. Rozonoer, Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning, Avtomatika i Telemekhanika, Vol. 25, No. 6, pp. 917-936, June, 1964.

[39] O. A. Bashkirov, E. M. Braverman and I. B. Muchnik, Potential Function Algorithms for Pattern Recognition Learning Machines, Avtomatika i Telemekhanika, Vol. 25, No. 5, pp. 629-631, May, 1964.

[40] Teuvo Kohonen, György Barna, and Ronald Chrisley, Statistical Pattern Recognition with Neural Networks: Benchmarking Studies, Vol. 1, pp. 61-68.

[41] Kevin J. Lang and Michael J. Witbrock, Learning to Tell Two Spirals Apart, Proceedings of Connectionist Models Summer School, 1988.

[42] Belur V. Dasarathy, Noising Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, pp. 67-72, January 1980.

[43] Geoffrey W. Gates, The Reduced Nearest Neighbor Rule, IEEE Transactions on Information Theory, pp. 431-433, May 1972.

[44] R. Paul Gorman and Terrence J. Sejnowski, Learned Classification of Sonar Targets Using a Massively Parallel Network, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 36, pp. 1135-1140, No. 7, July 1988.

[45] R. Paul Gorman, Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets, Neural Network, Vol. 1, pp. 75-89, 1988.

[46] Robert Detrano, Walter Steinbrunn, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H. Guppy, Stella Lee and Victor Froelicher, International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease, The American Journal of Cardiology, Vol. 64, pp. 304-309, August, 1989.

[47] PC MATLAB Version 4.2C. The Mathworks, Inc. Boston, Massachusetts, 1994.

[48] Accessible via anonymous FTP on ftp.cs.cmu.edu [128.2.206.173] in directory /afs/cs/project/connect/bench.

[49] Accessible via anonymous FTP on ftp.ics.uci.edu [128.195.1.1] in directory /pub/machine-learning-databases.

[50] Lloyd L. Allred, Personal Communication, 1996.

[51] Hong Jiang, Performance Analysis of Probabilistic Potential Function Neural Network, M.S. Thesis, The University of Toledo, Toledo, Ohio, December 1996.

Appendix 1 - Parameter Values for Neural Network Algorithms

Parameters presented in the following tables refer to variables employed in MATLAB Neural Network Toolbox simulation environment.

	Two-Spiral					IRIS			
PARAMETER	LVQ	RBF	MLP	PNN	PPFNN	LVQ	RBF	MLP	PPFNN
number of total iterations	1000		28000			1000		15000	
number of iterations for each presentation	1000		14000			1000		5000	
momentum constant			0.5					0.9	
bias time constant	0.9					0.9			
error goal			0.01			0.01		0.01	
learning rate	0.01		0.001					0.02	
number of hidden layer nodes	10	192	10, 10	79	70	20	120	100, 100	10
spread of RBFs		1.8					1.8		
Alpha				1.8	1.8				0.46
maximum error ratio			0.4					0.2	

	SONAR					VOWEL			
PARAMETER	LVQ	RBF	MLP	PNN	PPFNN	LVQ	RBF	MLP	PPFNN
number of total iterations	1000		10000			1000		5000	
number of iterations for each presentation	1000		5000			1000		5000	
momentum constant			0.0					0.0	
time constant	0.9					0.9			
learning rate	0.01		2.0			0.01		2.0	
error goal			0.02					0.02	
number of hidden layer nodes	300	192	24	68	66	50	150	24	100
spread of RBFs		1.8					1.8		
Alpha				5.3	4.4				9.0
maximum error ratio			1.06					0.2	

	WISCONSIN					CLEVELAND				
PARAMETER	LVQ	RBF	MLP	PNN	PPFNN	LVQ	RBF	MLP	PNN	PPFNN
number of total iterations	1000		5000			1000		5000		
number of iterations for each presentation	1000		5000			1000		5000		
momentum constant			0.0		40			0.5		
error goal			0.02			0.9				
bias time constant	0.9							0.1		
learning rate	0.01		2.0			0.01		0.001		
number of hidden layer nodes	45	150	24	11	5	65	201	10, 10	137	115
spread of RBFs		1.8					1.8			
Alpha				6.8	6.8				3.8	3.8
maximum error ratio			0.2					0.4		

	THYROID			
PARAMETER	LVQ	RBF	MLP	PPFNN
number of total iterations	1000		14000	
number of iterations for each presentation	1000		7000	
momentum constant			0.5	
bias time constant	0.9			
learning rate	0.01		0.001	
error goal			0.1	
number of hidden layer nodes	25	172	10, 10	12
spread of RBFs		1.8		
Alpha				0.0055
maximum error ratio			0.4	

Michael Wolfe
Report not available at time of publication.

A Scheduling Software Designed for Manufacturing Shop Floor

Norman D. Zhou
Associate Professor
Department of Technology

University of Wisconsin-Stout
Menomonie, WI

Final Report for:
Summer Research Extension Program

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Sacramento Air Logistics Center

December 1996

A SCHEDULING SOFTWARE DESIGNED FOR MANUFACTURING SHOP FLOOR

Norman D. Zhou
Associate Professor
Technology Department
University of Wisconsin-Stout

Abstract

Under manufacturing environment, the efficiency of our operations is the key to meet cost, schedule and performance commitments. To overcome the shortcomings of manual scheduling, a scheduling software SCHEDULER is designed. The SCHEDULER has optimization capability because its algorithm employs neural networks. The SCHEDULER is very user-friendly because it is windows-based and it has a Database entry format. The SCHEDULER is very powerful because it can deal with versatile scenarios. In this paper, the details of the SCHEDULER are presented.

A SCHEDULING SOFTWARE DESIGNED FOR MANUFACTURING SHOP FLOOR

Norman D. Zhou

Introduction

Manual scheduling has the following shortcomings:

1 People-dependent

The rules how to schedule are in scheduler's mind. The solution totally depends on him/her. If for some reason a scheduler asks his back-up to perform his task, everything will be somewhat different if not a mess because no two persons can schedule identically.

2 Non-optimal solution

There are always a few scheduling constraints, such as limited number of machines or operators. Because of doing it manually, a scheduler can only schedule orders according to priority number, then take care of those constraints. Human brain's capability is limited to play too many parameters at the same time. As a consequence, the schedule solution usually is only feasible and by no means of optimal.

3 Little simulation and analysis

It is necessary to have alternative schedules to make comparisons and to do analysis if a better schedule is desired. However it is not possible to manually simulate a situation and tie schedule, machine utilization, production backlog and earned hours together. Furthermore, this diversity of manual scheduling causes a serious gap between a useful algorithm and its practical application in the industry. The data format employed by different schedulers is not uniform so that it is not ready to meet the data entry requirement of a software.

In this paper, a windows-based user-friendly scheduling software is designed. Part I discusses the software's algorithm, which covers the discussion of problem and methodology. Part II describes the software's implementation, which includes the

description of installation, data source, dialog boxes, command bar buttons, scheduling outcome, file menu, edit menu, view menu, line menu, priority menu, calendar menu, and an example.

Discussion of problem---Statement of job-shop scheduling

Job-shop scheduling is a resource allocation problem. The resources are called machines, the tasks jobs. Each job may consist of several subjobs (referred to as operations) subject to some precedence restrictions. Job-shop scheduling is a classical operations research problem with numerous applications but very few practical solution approaches [Conway, et al., 1967]. Due to the large number of constraints the problem is known to be very hard in comparison with other combinatorial problems (e.g. TSP), so that even a good (not necessarily optimal) feasible solution (satisfying the constraints) is acceptable in most applications.

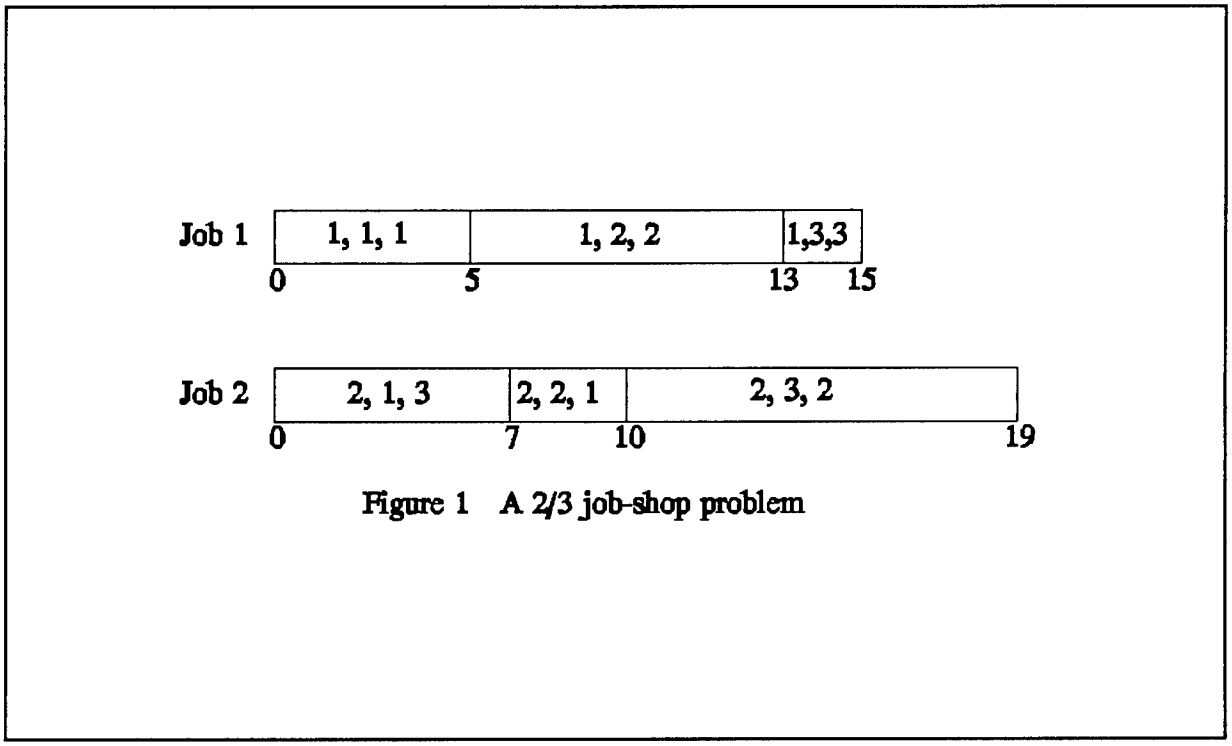


Figure 1 A 2/3 job-shop problem

Fig. 1 illustrates a 2/3 job-shop problem, i.e., two jobs, three machines, and three operations for each job, which is used as an example to illustrate the representation of the general job-shop problem (this example is borrowed from [Foo and Takefuji, 1988]).

Each job i consists of k_i operations. Each operation has three identifiers: i,j, and k, denoted as (k, j, k), where i represents the job number to which the operation belongs, j the sequence number of the operation, and k the number of the machine required to perform the operation. The length of each operation block in Fig. 1 is proportional to operation processing time, and the numbers underneath the block are used to indicate the completion time. A feasible schedule is given by the starting times of all operations so that the operations of each job will be performed in the required order and there will be no conflicts on each machine (i.e., operations do not overlap in time). Fig. 2 illustrates the solution form (the optimal schedule for the problem in Fig. 1). The operation blocks are rearranged into rows by machine numbers. The goal is to find a schedule to finish a set of jobs in the shortest time subject to constraints.

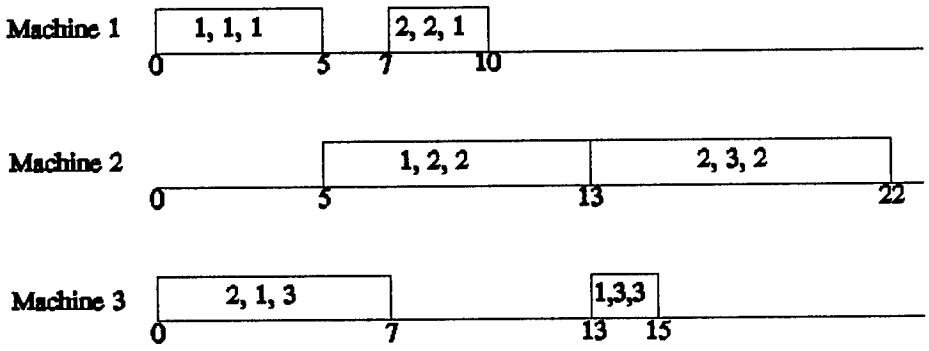


Figure 2 A 2/3 job-shop solution (optimal schedule)

The following presents a formal description of the job-shop scheduling problem. Let S_{ik} denote the starting time and t_{ik} the processing time for operation k of job i. The cost function will

be the sum of all the starting times of each job's last operation:

$$\sum_{i=1}^{n} S_{ik_i} ,$$

where n is the number of jobs and k_i the last operation of job i. The precedence constraint is given by the following inequality:

$$S_{ik} - S_{i,k+1} + t_{ik} <= 0, \quad 1 <= i <= n, \quad 1 <= k <= n - 1$$

assuming the number of jobs and the number of operations for each job are the same. This assumption is not restrictive and is used only to simplify the presentation. Note that the machine number does not appear with precedence constraints.

The condition that the starting time should be positive results in the constraint: $S_{i1} >= 0, \quad 1 <= i <= n.$

For any two operations (i, k) and (j, p) assigned to the same machine (here only the first two identifiers out of the three are needed to define an operation, where the first index denotes the job number and the second index the operation number), the following constraints need to be satisfied in order to avoid the overlap (in time) between these operations:

$S_{ik} - S_{jp} + t_{ik} <= 0$ if operation (i, k) is performed first

or

$S_{jp} - S_{ik} + t_{jp} <= 0$ if operation (j, p) is performed first. These are called disjunctive constraints because one or the other must hold.

Therefore the mathematical formulation for solving the job-shop scheduling is described as follows:

$$Minimize \sum_{i=1}^{n} S_{ik_i}$$

subject to the constraints

$$S_{ik} - S_{i,k+1} + t_{ik} <= 0,$$

$$S_{i1} >= 0,$$

$$S_{ik} - S_{jp} + t_{ik} <= 0 \quad or \quad S_{jp} - S_{ik} + t_{jp} <= 0.$$

Methodology---Scaling Neural Networks algorithm

Neural networks have been proposed as a computation model for solving a rich class of optimization problems. For example, N-queen, four-coloring and K-colorability, graph planarization , channel routing, knight's tour, spare allocation, tiling, and so on [Takefuji, 1992]. A large class of logical problems arising from real world situations can be formulated as optimization problems. Many of them are NP-complete, the time complexity functions of which are exponential [Garey and Johnson, 1979]. Thus neural networks provide a new approach to solve these complex combinatorial optimization problems through their intrinsic parallelism and heuristic nature.

Among these optimization problems, there is an important class of scheduling problems which many researchers have been addressing. For example, neural networks have been applied to job-shop scheduling [Foo and Takefuji 1988; Zhou and Cherkassky et al. 1990, 1991], dynamic load balancing [Oglesby and Mason, 1989], delivery truck scheduling [Davis et al., 1990], job sequencing [Fang et al., 1990], large-scale plant construction scheduling [Kobayashi and Nonaka, 1990], manufacturing task scheduling [Lo and Bavarian, 1991], and many other scheduling problems. Among these scheduling problems, job-shop scheduling is the most typical and classical. So far there are three different neural networks have been applied to it and comparisons have been made with the conventional heuristics of priority dispatching rules.

The representation scheme for mapping the job-shop scheduling problem onto the scaling neural network is described below. The starting time of each operation is represented as the state of the elementary neural processor. Hence, the number of neurons is the same as the number of the starting time variables of all operations.

According to the job-shop scheduling problem formulation in the previous section the computational energy function is constructed as follows:

where the first summation index (from i=1 to n) is omitted for

$$E = S_{ik_i} + \sum_{k=1}^{k_i-1} H_1 * F_1 (S_{ik} - S_{i,k+1} + t_{ik}) + H_2 * F_1 (-S_{i1}) +$$

$$\sum_{1 \le k \le k_i, 1 \le p \le k_j, 1 \le i < j \le n, m(i,k) = m(j,p)} H_3 F_2 (S_{ik} - S_{jp} + t_{ik}) F_2 (S_{jp} - S_{ik} + t_{jp}),$$

simplicity; the first term is the problem cost function, the second term is for the precedence constraint, the third term is for the constraint that the starting time needs to be positive, and the last term is to resolve the disjunctive constraint; H_i is a large positive constant; $F_1(x) = e^{bx} - bx$ and $F_2(x) = e^{bx} - 1$ when $x > 0$, they are equal to 0 when $x <= 0$; b is a constant depending on the characteristic of the diode.

Thus, the minimum of the computational energy function E corresponds to the shortest total completion time of all jobs in a feasible solution, because when all the constraints are satisfied, the corresponding constraint terms are equal to zero.

Results---Development of a scheduling software

To overcome the gap between a useful algorithm and the practical application in real life, a windows-based software SCHEDULER is developed. The SCHEDULER provides tools to help schedule manufacturing jobs (or tasks) with optimal solutions, generate Gantt bar solutions with time scale, and analyze machine utilization.

The SCHEDULER can handle many jobs (or tasks) up to 100 jobs, each job can have one operation or multiple operations. The number of production lines is designed three lines, each line can have fifteen different machines.

The SCHEDULER schedules jobs according to the rules set up by the user, for example, due dates, and it will satisfy constraints, such as the constraint of secondary machines.

In the following sections, the details of the SCHEDULER are presented.

1. Features

- Dialog boxes

  Dialog boxes are employed for all the data entry.

- Gantt chart

The schedule solution is presented by the Gantt chart with the time scale at the top which can be zoomed in/out to show different shifts/different days.

- Windows-based

The SCHEDULER runs under Microsoft Windows 3.1 or above with Menu system and toolbars for all the commands.

- Scroll bar

The Gantt chart solution is scrollable. One can view the solution part which is not displayed in the current window by using the scroll bar.

2. Functionality

- Database

There are five tables: Job table, Line table, Machine table, Operation table, Secondary Machine table. Double clicking the operation bar in the Gantt chart solution will show some key information in the Job and Operation tables.

- Print and Print preview

The whole Gantt chart solution can be previewed and printed in multiple pages.

- Filter dialog boxes and Filter button

The filter dialog box is used to specify the part of the information you want to edit or view. The Filter button is used if one wants to view only some specific records.

- Make Job Set 1 and Job Set 2

Jobs can be divided into two sets according to the due dates set up in the Make Job Set ½ menu to increase the machine utilization.

- Browse Job Set 1 and Job Set 2

View/Job Set 1 and 2 is used to browse the schedule solution for Job Set 1 and 2 respectively in the Gantt chart.

- Zoom-in/out

Zoom-in is used to enlarge the Gantt chart to see more detailed information. Zoom-out is used to shrink the Gantt chart to see more of the whole chart especially for the case when the chart is beyond one window screen.

- Priority of heuristic rules

Jobs can be scheduled according to due dates, machine size, or other parameters such as material color, mold size if needed. The priority of these heuristic rules can also be set up.

- Set up working/non-working days

Calendar menu generates a Calendar dialog box,, in which one can set working or non-working days. The SCHEDULER

schedules jobs according to the working days set up in its calendar.

- Delay

Schedule/Delay menu is used to delay all the jobs in process or waiting for a specific period of time in case of machine down or some other reason.

3. System requirements

- An IBM pc or 100% compatible that runs Microsoft Windows 3.1 or above.
- The Microsoft Open DataBase Connectivity (ODBC) installed. The desired database for example FoxPro or Oracle can be chosen.
- A color printer needed to generate scheduling charts.

4. Installation

Either use the setup program or simply copy the SCHEDULE.EXE to c:\schedule.

5. Data source

The ODBC data source has to be set up before running the program. The following procedure can be followed:

1) Copy the SCHEDULER's data files (*.dbf) to the hard disk in a sub-directory c:\schedule\datasrc\.
2) Start the ODBC Manager in the Control Panel by double clicking its icon.
3) In the Data Source dialog box, press Add.. and in the Installed ODBC Drivers list, select the desired Files for example FoxPro (*.dbf) and press OK. The ODBC FoxPro Setup dialog box appears.
4) In the Data Source Name edit box, type in Scheduler. In the description edit box, type in Manufacturing Scheduler.

5) Press the Select Directory button to select the directory to where the *.dbf files have been copied (suggested above as c:\schedule\datasrc\).
6) Press OK to close the ODBC FoxPro Setup dialog box. Press Close to close the Data Source dialog box.

6. Dialog boxes

There are five tables in the database as follows:

A Job table
B Line table
C Machine table
D Operation table
E Secondary Machine table

Each table has its own fields. For example, for Job table, there are the following fields: Job ID, Order ID, Part No., Customer, Description, Quantity, Processing time, Total time, Next job, Due date, Release date, Material, Material availability, Line ID, Operation No., Comment, and a status field to indicate the job completed, in the process or waiting. The details are shown in the figures from three to seven.

All the above tables have dialog boxes described in detail below.

- Viewing dialog boxes

Viewing dialog boxes are used to view the information in the database. Use >>, <<, > and < buttons to navigate through the database. Use the Filter button to view specified records (see below for detailed information).

- Editing dialog boxes

Editing dialog boxes are used to edit the information in the database. In addition to the above buttons described, there are Remove button to delete the current record that is displayed and New button to add a new record. The following

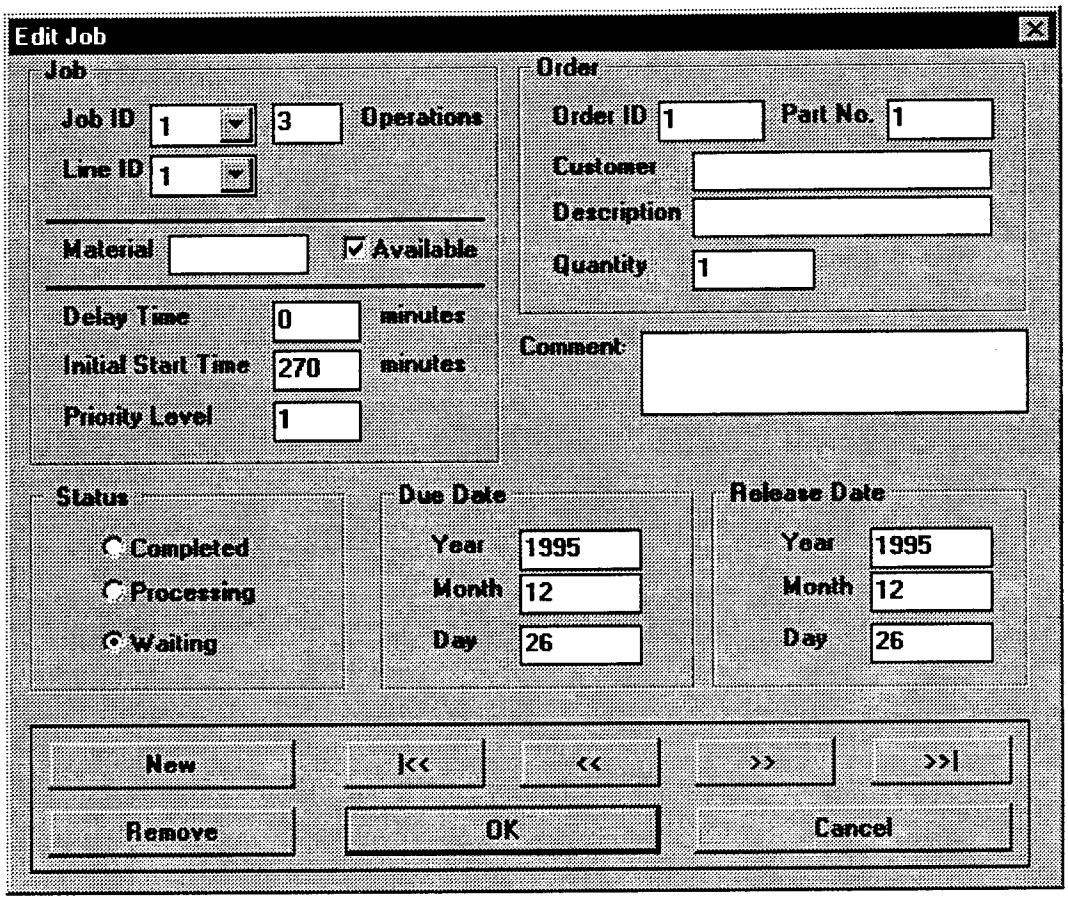


Figure 3 Job Table Dialog Box

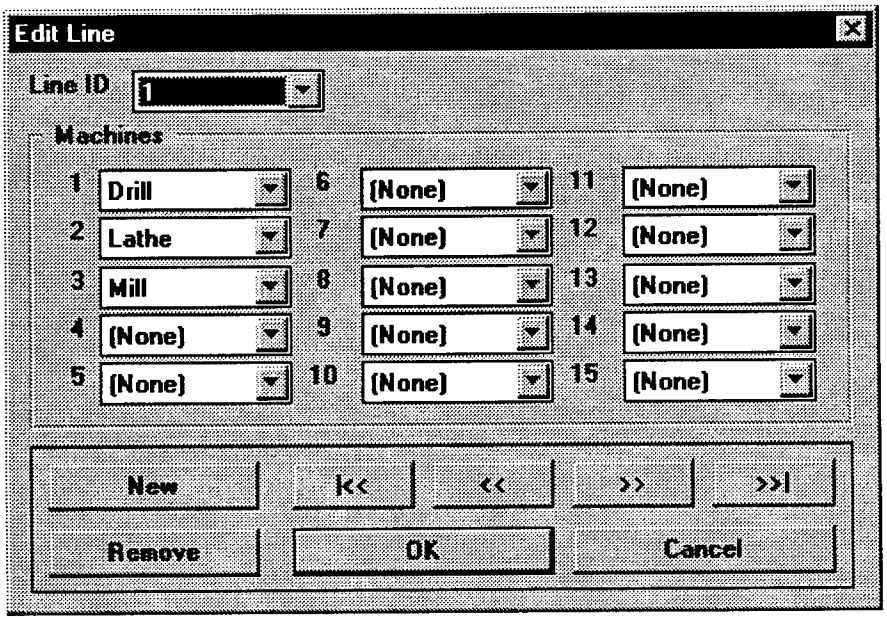


Figure 4 Line Table Dialog Box

figures from three to seven are the five dialog boxes for the above five tables.

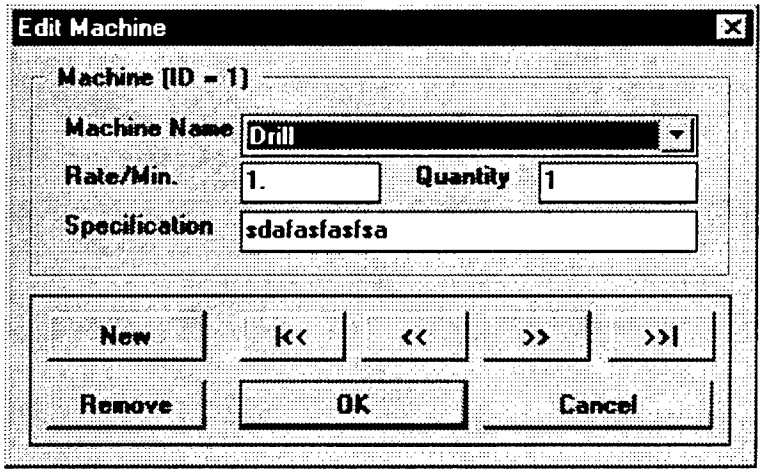


Figure 5 Machine Table Dialog Box

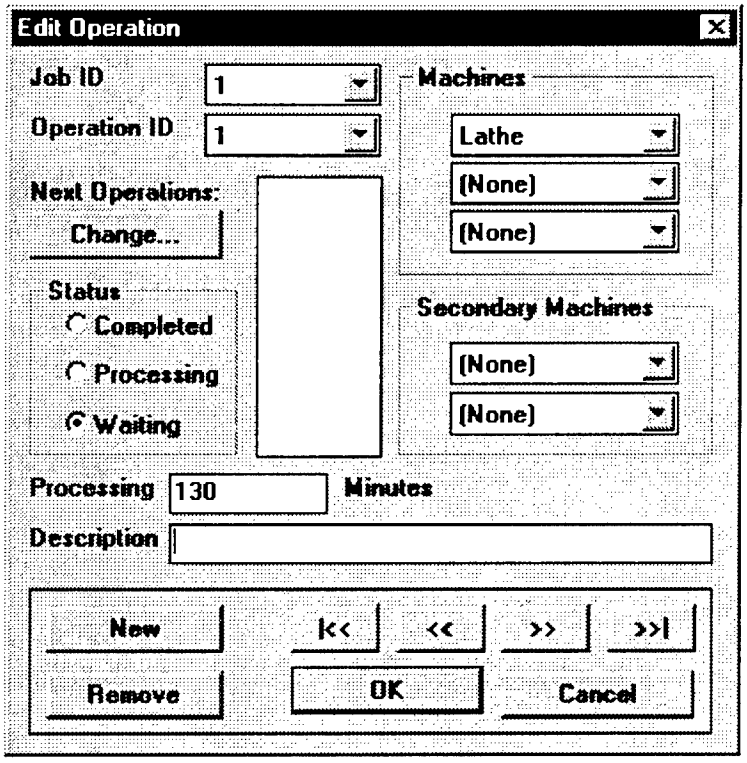


Figure 6 Operation Table Dialog Box

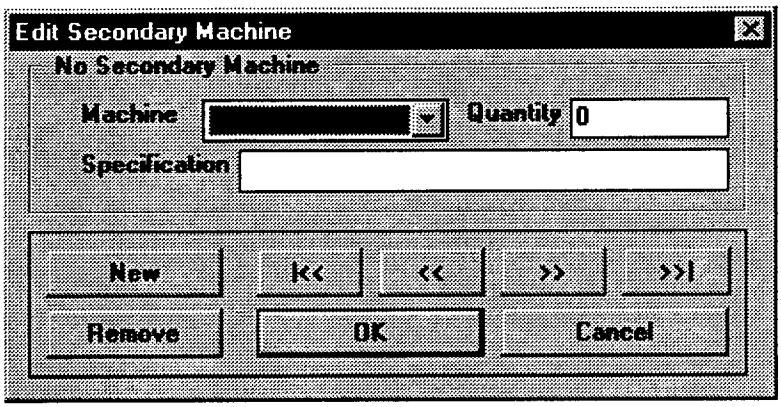


Figure 7 Secondary Machine Table Dialog Box

- Filtering dialog boxes

Filtering dialog boxes are used to specify the part of the information you want to view or edit. One can give some values in the filtering dialog boxes. Only those records that have such values can be viewed or edited.

- Calendar dialog box (Figure 8)

Figure 8 Calendar Dialog Box

The Calendar dialog box is not associated with the above tables. It is used to set working and non-working days. All days that are either Saturday or Sunday are preset to non-working days. The other days are all working days. To toggle a working day to a non-working day, just click on the button of that day, vice versa. The color of that day will change accordingly to indicate the new status.

Use the >>, <<, < and > buttons to navigate through the calendar. Choose OK to accept the changes one has made since the dialog box opened. Press Remove Before to delete the months before the current month from the calendar. Press New to append a new month to the calendar.

7. Command bar buttons

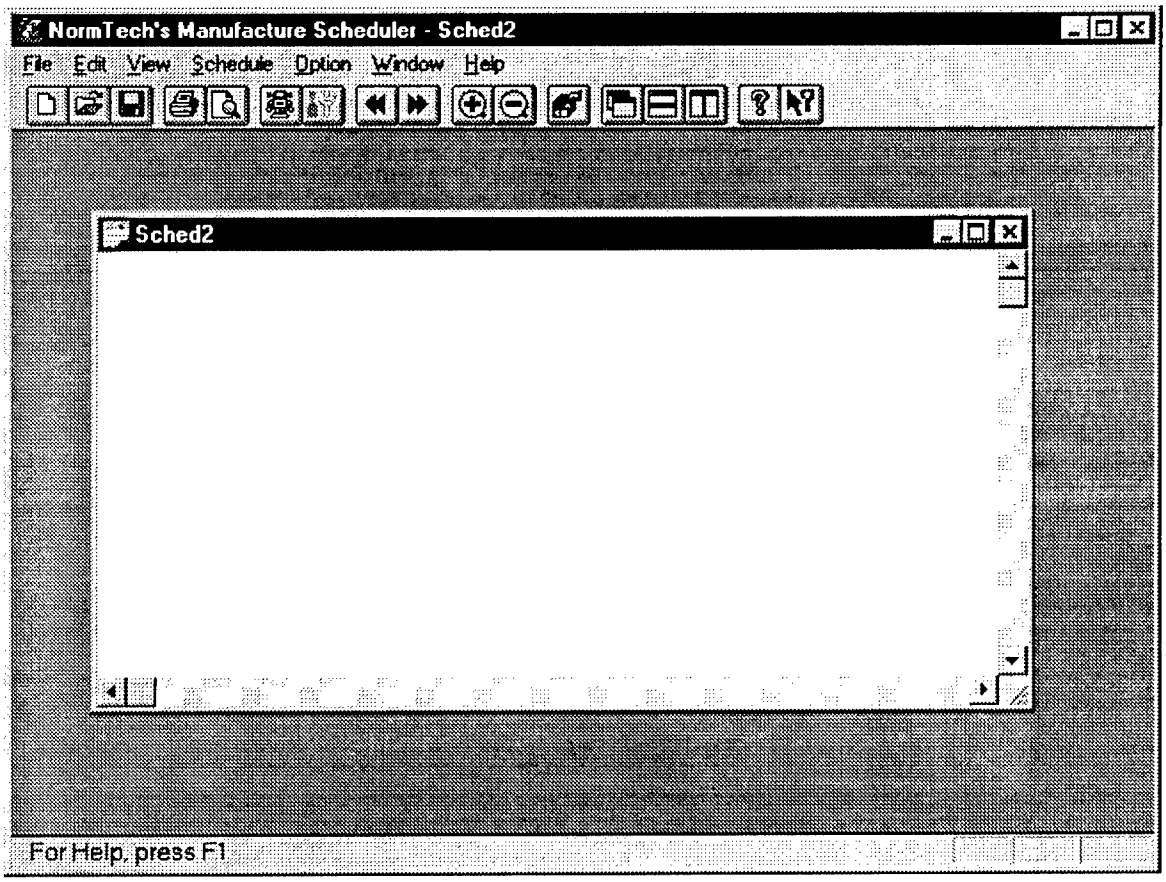


Figure 9 Command Bar Buttons

The Command Bar Buttons are shown below.

Every toolbar button corresponds to a menu command. The mapping of buttons and menu commands is listed in the order of the buttons from left to right as they appear in the Windows.

Button	**Menu Command**
New	File/New
Open	File/Open
Save	File/Save
Print	File/Print
Preview	File/Print Preview
Edit Job	Edit/Job
Edit Operation	Edit/Operation
Previous Day	View/From Previous Day
Next Day	View/From Next Day
Zoom In	View/Zoom In
Zoom Out	View/Zoom Out
Run Set 1	Schedule/Run Job Set 1

Cascade, Tile Horizontal, Tile Vertical, About and Help are common to all windows-based software.

8. Scheduling outcome

The SCHEDULER shows its outcome, i.e., an optimal schedule solution, in its window's client area (Figure 10). All operations (or jobs) are drawn as 3-D colored bars whose lengths represent the processing times of operations. The position of the bars represents the start time of operations. Underneath the schedule chart are the hint bars that show the mapping of the Job-Operation Numbers and the bar colors.

The detailed information of an operation (or a job) can be viewed by double clicking the color bar of that operation (or a job) either in the schedule chart or in the hint area. If one does so, a dialog box is popped up to display the detailed information of that operation (or that job).

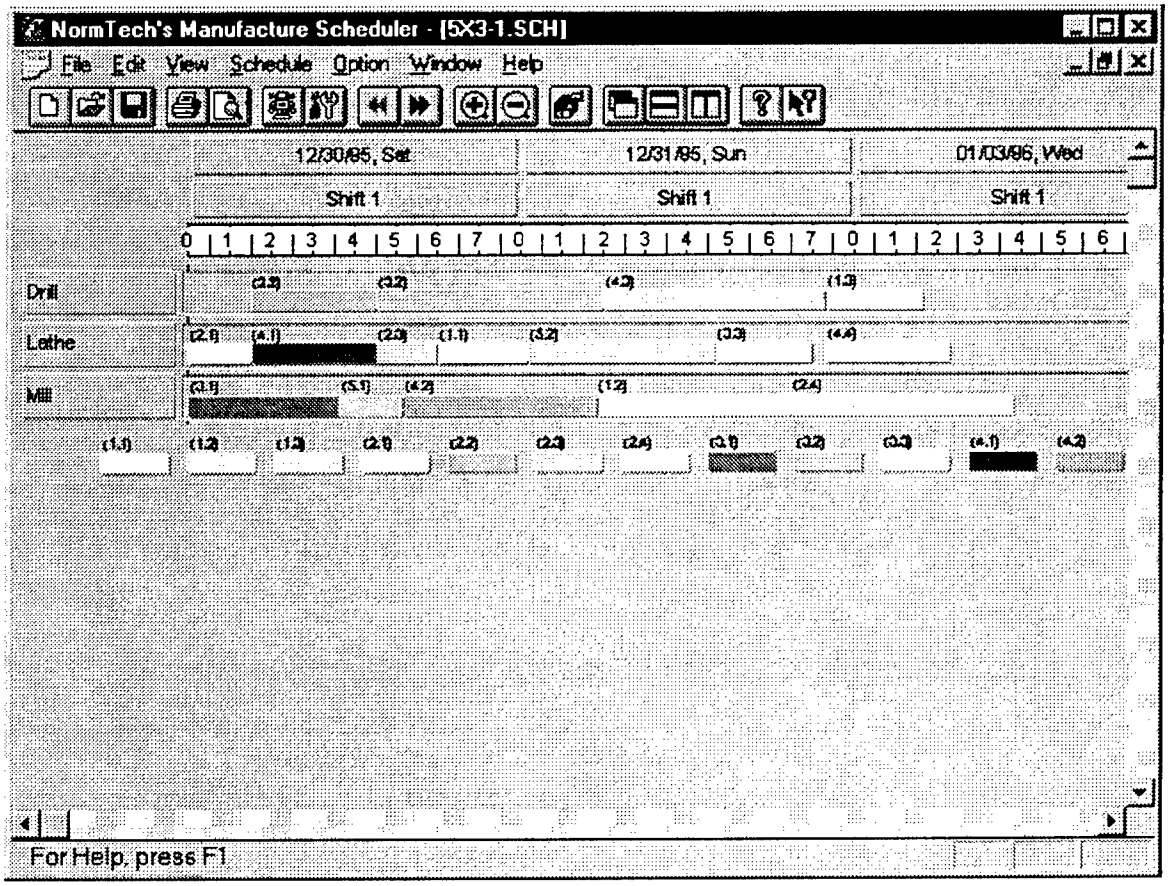


Figure 10 An Optimum Schedule for a 5/3 Job-shop Problem Generated by the software SCHEDULER

9. ORACLE database

A database could be employed as an interface between the scheduling software SCHEDULER and schedulers to make data entry and communication a lot easier. The ORACLE database may be used because it is the most powerful database. With ODBC in the Visual C++, ORACLE database can be integrated into the scheduling software to be a universal software agent applied to versatile situations.

10. Special concerns for manufacturing environment

(a) Machine utilization

In practice, one of the goals is to increase the utilization of machines in addition to achieving a shorter completion time to meet customer's due date.

To achieve this goal, jobs will be put into two sets according to due date: Job Set 1 and Job Set 2. Job Set 1 is run by the SNN algorithm first to generate an optimal schedule. However, there are still idle times of machines. Then Job Set 2 will be run to fill in these gaps as many as possible. One SNN algorithm is run twice. The only difference is that the starting times of the operations in Job Set 1 are fixed in the second run.

(b) Parallel (identical) machines

There are identical machines in practice. For example, there are a few mills and a few lathes with the same specifications. The SNN algorithm is improved to handle such a situation easily with the same goal in mind: to minimize the completion time of jobs.

(c) Operation-split case

The original SNN like the other neural networks or conventional algorithms assumes that every operation only has one subsequent operation. In reality, it is possible to have a few operations not start until one specific operation is completed. On the job-shop problem chart, one operation leads not only to one but to several operations. The SNN is improved to work gracefully with this operations-split situation by modifying a constraint term in the energy function of the SNN algorithm.

(d) Operations-combined case

Of course, operations-combined problems happen very often in practice. For example, operations are combined when two parts are assembled together.

This situation is equivalent to having several pre-operations instead of only one pre-operation for every operation in the job-shop problem chart. The SNN is improved to solve this situation neatly.

Conclusion

The scheduling software SCHEDULER employs the improved SNN algorithm, which can handle versatile practical manufacturing situations. The SCHEDULER is windows-based and very user-friendly. The economic impact is apparent because it offers optimal schedule solutions to increase the efficiency of production.

References

[1] Conway R.W., Maxwell W.L., Miller L.W. (1967), Theory of Scheduling, Reading, MA: Addison-Wesley.

[2] Davis D.L., Gillenwater E.L., Johnson J.D. (1990), "An artificial neural systems framework for delivery truck scheduling," *IEEE Comput. Soc. Press*, Los Alamitos, CA, USA, Vol. 3, pp. 327-333.

[3] Fang L., Wilson W.H., Li T. (1990), "A neural network for job sequencing," *Proc. Conf. Parallel Processing in Neural Systems and Computers*, Elsevier Science Publ. B.V., pp. 253-256.

[4] Foo Y.P.S., Takefuji Y. (1988), "Stochastic neural networks for solving job-shop scheduling," *Proc. IEEE IJCNN 88*, pp. 275-290.

[5] Foo Y.P.S., Takefuji Y. (1988), "Integer-linear programming neural networks for job-shop scheduling," *Proc. IEEE IJCNN 88*, pp. 342-348.

[6] Garey M.R., Johnson D.S. (1979), Computers and intractability: a guide to the theory of NP-completeness, San Francisco, W.H. Freeman and Company.

[7] Kobayashi Y., Nonaka H. (1990), "Application of neural networks to schedule integration in plant engineering," *Proc. International Neural Network Conference (INNC 90)*, Paris, France, Vol. 1, pp. 287-290.

[8] Lo Z.P., Bavarian B., "A modified Hopfield neural network for optimization of manufacturing task

scheduling," *Robotics and Manufacturing*, ASME Press, Vol. 3, 1991, pp. 1001-1006.

[9] Oglesby J., Mason J.S. (1989), "Dynamic scheduling for feed-forward neural nets using transputers," *Proc. First IEE Int. Conf. on Artificial Neural Networks (ICANN 89)*, 16-18 Oct. 1989, London, UK, pp.257-260.

[10] Takefuji Y. (1992), Neural network parallel computing, Norwell, Massachusetts, Kluwer Academic Publishers.

[11] Zhou D.N. (1992), A Scaling Neural Network for Job-shop Scheduling Problems, UMI, A Bell & Howell Information Company, 1992.

[12] Zhou D.N., Cherkassky V., Baldwin T.R., Olson D.E. (1991), "A neural network approach to job-shop scheduling," *IEEE Trans. on Neural Networks*, vol. 2, no. 1, pp. 175-179.

[13] Zhou D.N., Cherkassky V., Baldwin T.R., Hong D.W. (1990), "Scaling neural network for job-shop scheduling," *Proc. IJCNN*, vol. 3, pp. 889-894.

Adaptive Control of Active Magnetic Bearings Under Unknown Static Load Change and Rotor Imbalance

James J. Carroll
Assistant Professor
Department of Electrical & Computer Engineering

Clarkson University
Potsdam, NY 13699

Final Report for:
Summer Research Extension Program

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

and

Warner Robins Air Logistics Center

December 1996

ADAPTIVE CONTROL OF ACTIVE MAGNETIC BEARINGS UNDER UNKNOWN STATIC LOAD CHANGE AND ROTOR IMBALANCE

James J. Carroll

Assistant Professor

Department of Electrical & Computer Engineering

Clarkson University

Abstract

An adaptive controller based on an integrator backstepping technique is proposed for active magnetic bearing (AMB) systems suffering unknown static load changes and rotor imbalances, i.e., two of the most common and challenging forms of disturbances in practical AMB systems. The proposed controller ensures that the rotor of the AMB system is asymptotically stabilized to its geometric axis, and as such, offers substantial performance improvements over prior results that could only stabilize the AMB system to its principal axis of inertia, e.g., this allows the steady-state stiffness of the AMB system to be arbitrarily adjusted. A side benefit of the proposed adaptive control scheme is that estimates of the unknown AMB system parameters related to the static load and rotor unbalance convergence to their actual values, and thus, can be used to provide real-time estimates of these difficult to measure parameters. This is, to the author's knowledge, a first in the literature for such Lyapunov-like controllers. Simulation results are presented as a first step in the verification of the proposed controller performance.

ADAPTIVE CONTROL OF ACTIVE MAGNETIC BEARINGS UNDER UNKNOWN STATIC LOAD CHANGE AND ROTOR IMBALANCE

James J. Carroll

I. Introduction

Magnetic bearings and related suspension systems offer many advantages over their conventional counterparts and research into these systems dates back to early last century. The reason for this attention is that active magnetic bearing (AMB) systems have a wide variety of potential applications from the mundane, e.g., HVAC compressor systems, to the exotic, e.g., gyros used in the inertial guidance systems of spacecraft, due to their high-performance operation which is nearly friction-free, lubricant-free and wear-free. For details on various AMB applications, the interested reader is referred to [7], [12], [13] and [14]. Stability has been a central issue associated magnetic bearing applications since early realizations that totally passive magnetic bearing systems are inherently unstable. Active control of magnetic bearing systems has proven to be the most effective way to solve the bearing stability problem, and as a result, AMB control systems have drawn an increasing amount of attention.

As noted above, the regulation of an AMB rotor at the geometric center of its stator requires the associated electromagnetic forces to be actively controlled, i.e., in response to the motion of the suspended rotor in the air gap. Currently, 90% of the industrial AMB systems use simple PID control laws [7] to achieve this control objective and generally these controllers are tuned in an ad-hoc fashion, i.e., based on trial and error without any mathematical model for guidance [10]. Since the control algorithm plays a crucial role in the operation of any AMB system, various advanced AMB control strategies have been proposed to improve system performance. Recently proposed techniques include: backstepping control with exact modeling knowledge [1], optimal control [9],[20], sliding mode control [10],[11],[15],[16],[19], Q-parameterization type control [17],[25], μ-synthesis type control [21] and loop shaping based robust control [23].

AMB controllers generally have one of two goals: (1) to reduce the total hardware cost by

eliminating need for, or requirements on, some components, or (2) to increase system robustness by incorporating modern digital signal processor (DSP) technology to realize more sophisticated control algorithms [7],[8]. In the former case, a self-sensing bearing configuration is proposed in [18] to eliminate the need for air-gap sensors. In the later case, a successful application of DSP's in the control of an AMB system for a milling spindle is reported in [7], which points to the future trends in AMB research. With increasing application of DSP's, it is predicted that current problems with AMB systems, such as compensation of sudden static load change, unbalance, or other disturbances, can be met by more sophisticated, model-based controllers. While certain AMB properties, like stiffness to static load change, can be increased drastically by adding integral control terms to standard controllers [7], simple PID controllers cannot handle complex disturbances like rotor imbalance.

There are two common ways to deal with rotor imbalance. The first is to produce an active balancing force which compensates for the disturbance force caused by rotor imbalance and causes the rotor to rotate about its geometric axis. The second method is to produce a signal which cancels the component of the air-gap sensor signal caused by the rotor imbalance. This second method results in rotation about the rotor's principal axis of inertia, as discussed in [25] using Q-parameterization theory. An auto-balancing adaptive control is also proposed in [8] to stabilize the unbalanced rotor about its principal axis of inertia. These control strategies use linear control arguments and the designed controllers operate successfully only in a specified frequency range (rotor speed). A drawback of the auto-balancing technique is that it affects the positioning accuracy of the rotor and the air-gap needs to be sufficiently large. Furthermore, for some AMB applications, such as compressors for vacuum pumps or fluid supplies, the rotor is inherently unbalanced by design and it is essential in these applications that the rotor rotate about its geometric axis, rather than its principal axis of inertia.

There are often other limitations associated with linear control designs. As noted in [22], when an AMB system is locally stabilized around an operating point using linear control methods, nonlinear oscillations can occur at certain critical rotor speeds. In addition, some of the linear

methods [7] designed to treat AMB unbalance based on locally-linearized models suppress other desired control actions, leading once again to system instability. Given this, it is reasonable to consider nonlinear control techniques for AMB systems. While nonlinear compensation of an AMB system was successfully implemented using analog circuits in [24], it seems more likely that future implementations of nonlinear controllers will rely on cheap DSP-based hardware.

In this paper, a fully nonlinear AMB model is used to develop an adaptive controller (based on an integrator backstepping technique) for AMB systems which suffer unknown static load change and rotor imbalance, i.e., the two most common and challenging disturbances in AMB systems. The proposed controller asymptotically stabilizes the rotor to its geometric axis (in contrast to the those results that yield an unbalanced AMB rotor stabilized to its principal axis of inertia). The proposed control scheme is capable of stabilizing the AMB rotor about its geometric axis despite static load changes. This implies that the steady-state stiffness of the closed-loop AMB system can be made arbitrarily large. Furthermore, the proposed Lyapunov-type adaptive control provides real-time estimates of the unknown system parameters related to static load change and rotor imbalance. It is shown that these estimates convergence to their true values.

The paper is organized as follows. The system model is described in Section II. The voltage level adaptive control is designed in Section III. The proof of the parameter convergence is given in Section IV. Simulation results are presented in Section V and the paper is concluded in Section VI.

II. System Model

For control purposes, we consider a horizontally positioned shaft which has an AMB system attached at the end, as shown in Figure 1. By neglecting gyroscopic effects, the dynamics of the system along the vertical axis and the horizontal axis can be decoupled. This simplification seems reasonable since in practice [7],[8], most industrial AMB systems are controlled by "local" controllers, i.e., controller for each axis is designed and implemented independently. Furthermore, according to [7], the minimal increase in performance associated with fully-coupled feedback does

not justify the increased expense in control complexity. Therefore, we shall describe our control strategy by considering only one axis, i.e., the vertical axis of the single-end magnetic bearing, though the design is readily extendible to fully-coupled systems.

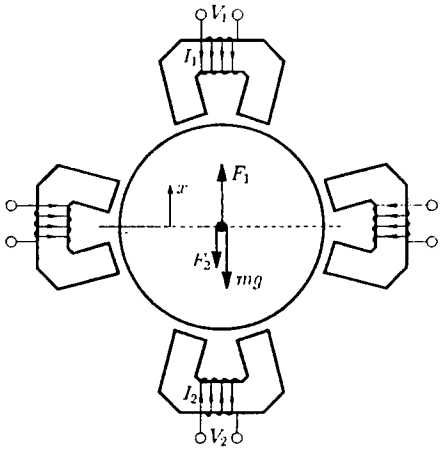


Figure 1. Cross section of an AMB system.

As is common in the literature [12]-[16], we assume that all of the electromagnets are identical and neglect the associated magnetic hysteresis, saturation, eddy current, and back EMF effects. The resulting electromechanical dynamics of the magnetic bearing in the vertical axis are

$$m\ddot{x} = F_1 - F_2 - mg + T_d, \tag{1}$$

$$F_1 = k\frac{I_1^2}{(w - x + \delta)^2}, \tag{2}$$

$$F_2 = k\frac{I_2^2}{(w + x + \delta)^2}, \tag{3}$$

$$L\dot{I}_1 + RI_1 = V_1, \tag{4}$$

$$L\dot{I}_2 + RI_2 = V_2, \tag{5}$$

where g is the gravity; w is the nominal air-gap length; x is the displacement of the rotor away from the geometric center and is constrained in the air-gap, i.e., $x \leq w$; F_1 and F_2 are the magnetic forces produced by the top electromagnet and the bottom electromagnet in the vertical axis, respectively; k is a positive force coefficient; δ is a positive constant representing the effect of the permeability of the magnetic material; L and R are the inductance and resistance respectively,

associated with each coil; V_1 and V_2 are the control voltages; T_d is the disturbance including static load change and unbalance and can be described as

$$T_d = c_1 + c_2' \sin(\Omega t + \theta_0), \tag{6}$$

where c_1 is the unknown static load change and the second term represents the disturbance force in the vertical axis caused by unbalance with c_2' and θ_0 being unknown constants and Ω being a known constant rotor speed. Note, c_2' is proportional to Ω squared and rotor eccentricity, i.e., the distance between the geometric center and mass center of the rotor.

III. Adaptive Control Design

Assuming full state measurements, i.e., rotor position, velocity and coil currents, a voltage-level adaptive control for the AMB system described by (1) through (6) will be designed. To facilitate the control design, the disturbance of (6) is written in an equivalent form, as shown below[8]

$$T_d = c^T \zeta, \tag{7}$$

where $c = [c_1 \ c_2 \ c_3]^T$ is unknown constant vector, and

$$\zeta = [1 \ \sin(\Omega t) \ \cos(\Omega t)]^T. \tag{8}$$

A filtered displacement variable r is defined as shown

$$r = \dot{x} + \alpha x, \tag{9}$$

where α is a positive control gain[6].

Substituting (7), (9), (2), and (3) into (1), we obtain the dynamics of the mechanical subsystem expressed in terms of the filtered displacement

$$m\dot{r} = \alpha m\dot{x} + k\frac{I_1^2}{(w - x + \delta)^2} - k\frac{I_2^2}{(w + x + \delta)^2} - mg + c^T\zeta. \tag{10}$$

Adding and subtracting the fictitious currents I_{1c} and I_{2c} respectively to I_1 and I_2 in (10), we have

$$m\dot{r} = \alpha m\dot{x} - mg + c^T\zeta + k\frac{I_{1c}^2}{(w - x + \delta)^2} - k\frac{I_{2c}^2}{(w + x + \delta)^2}$$

$$-k\frac{I_1+I_{1c}}{(w-x+\delta)^2}\eta_1+k\frac{I_2+I_{2c}}{(w+x+\delta)^2}\eta_2, \tag{11}$$

where

$$\eta_1 = I_{1c}-I_1, \tag{12}$$

$$\eta_2 = I_{2c}-I_2. \tag{13}$$

We define I_{1c} and I_{2c} as follows

$$I_{1c} = [\frac{(w-x+\delta)^2}{k}\cdot|k_v r+\alpha m\dot{x}-mg+\hat{c}^T\zeta|_\sigma]^{1/2}, \tag{14}$$

$$I_{2c} = [\frac{(w+x+\delta)^2}{k}(k_v r+\alpha m\dot{x}-mg+\hat{c}^T\zeta$$

$$+|k_v r+\alpha m\dot{x}-mg+\hat{c}^T\zeta|_\sigma)]^{1/2}, \tag{15}$$

where k_v is a positive control gain; $\hat{c}^T$ is the estimation of c^T; and the σ-norm is defined as

$$|z|_\sigma = \sqrt{z^2+\sigma}, \tag{16}$$

with σ being a positive constant. Note, the value of σ will determine the steady state currents in the electromagnets. Substituting (14) and (15) into (11) yields

$$m\dot{r} = -k_v r+\tilde{c}^T\zeta+A\eta_1+B\eta_2, \tag{17}$$

where $\tilde{c}^T = c^T-\hat{c}^T$, i.e., the parametric estimation error, and

$$A = -k\frac{I_1+I_{1c}}{(w-x+\delta)^2}, \tag{18}$$

$$B = k\frac{I_2+I_{2c}}{(w+x+\delta)^2}. \tag{19}$$

Differentiating (12) and (13) yields

$$\dot{\eta}_1 = \dot{I}_{1c}-\dot{I}_1, \tag{20}$$

$$\dot{\eta}_2 = \dot{I}_{2c}-\dot{I}_2. \tag{21}$$

From (14) and (15), we can obtain

$$\dot{I}_{1c} = \frac{1}{2kI_{1c}}[-2(w-x+\delta)\dot{x}y+(w-x+\delta)^2\dot{y}], \tag{22}$$

$$\dot{I}_{2c} = \frac{1}{2kI_{2c}}[2(w+x+\delta)\dot{x}(s+y)+$$

$$+(w+x+\delta)^2(\dot{s}+\dot{y})], \tag{23}$$

where

$$y = \sqrt{s^2+\sigma}, \tag{24}$$

$$s = \alpha m\dot{x} - mg + k_v r + \hat{c}^T\zeta, \tag{25}$$

$$\dot{y} = \frac{s}{y}(\alpha m\ddot{x} + k_v\dot{r} + \frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta}), \tag{26}$$

$$\dot{s} = \alpha m\ddot{x} + k_v\dot{r} + \frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta}. \tag{27}$$

From (20) through (27), (4) and (5), we now obtain

$$L\dot{\eta}_1 = \frac{L}{2kI_{1c}}[-2(w-x+\delta)\dot{x}y + (w-x+\delta)^2\frac{s}{y}(\alpha m\ddot{x}$$

$$+k_v\dot{r} + \frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta})] + RI_1 - V_1, \tag{28}$$

$$L\dot{\eta}_2 = \frac{L}{2kI_{2c}}[2(w+x+\delta)\dot{x}(s+y) + (w+x+\delta)^2(1+\frac{s}{y})\cdot$$

$$\cdot(\alpha m\ddot{x} + k_v\dot{r} + \frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta})] + RI_2 - V_2. \tag{29}$$

To specify the control voltages V_1, V_2, and the update law of the parameter $\hat{c}$ (i.e., $\frac{d}{dt}\{\hat{c}\}$), a Lyapunov-like function [2] is defined as shown

$$\Lambda = \frac{1}{2}mr^2 + \frac{1}{2}L\eta_1^2 + \frac{1}{2}L\eta_2^2 + \frac{1}{2}\tilde{c}^T K_c^{-1}\tilde{c}, \tag{30}$$

where K_c is a 3×3 diagonal gain matrix with positive elements k_{c1}, k_{c2}, and k_{c3}. Differentiating (30) with respect to time yields

$$\dot{\Lambda} = rm\dot{r} + \eta_1 L\dot{\eta}_1 + \eta_2 L\dot{\eta}_2 - \tilde{c}^T K_c^{-1}\frac{d}{dt}\{\hat{c}\}. \tag{31}$$

After substituting the state trajectories of (17), (28) and (29) into (31), the control voltages V_1, V_2, and the parameter update law $\frac{d}{dt}\{\hat{c}\}$ will be specified such that $\dot{\Lambda}$ becomes

$$\dot{\Lambda} = -k_v r^2 - \Gamma_1\eta_1^2 - \Gamma_2\eta_2^2, \tag{32}$$

where Γ_1 and Γ_2 are positive control gains. Given this, the control voltages are found to be

$$V_1 = \frac{L}{2kI_{1c}}\{-2(w-x+\delta)\dot{x}y + (w-x+\delta)^2\frac{s}{y}\cdot$$

$$\cdot[(\frac{k_v}{m}+\alpha)(-k_v r + A\eta_1 + B\eta_2) - \alpha^2 m\dot{x}$$

$$+\frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta}]\} + \Gamma_1\eta_1 + RI_1 + A\cdot r, \tag{33}$$

$$V_2 = \frac{L}{2kI_{2c}}\{2(w+x+\delta)\dot{x}(s+y) + (w+x+\delta)^2(1+\frac{s}{y})\cdot$$

$$\cdot[(\frac{k_v}{m}+\alpha)(-k_v r + A\eta_1 + B\eta_2) - \alpha^2 m\dot{x}$$

$$+\frac{d}{dt}\{\hat{c}^T\}\zeta + \hat{c}^T\dot{\zeta}]\} + \Gamma_2\eta_2 + RI_2 + B\cdot r, \tag{34}$$

where $\dot{\zeta}$ is obtained from (8) as shown

$$\dot{\zeta} = [0\ \Omega\cos(\Omega t)\ \text{-}\Omega\sin(\Omega t)]^T, \tag{35}$$

and the parameter update law is specified as

$$\frac{d}{dt}\{\hat{c}\} = K_c[r + \frac{L}{2kI_{1c}}(w-x+\delta)^2\frac{s}{y}(\frac{k_v}{m}+\alpha)\eta_1$$

$$+\frac{L}{2kI_{2c}}(w+x+\delta)^2(1+\frac{s}{y})(\frac{k_v}{m}+\alpha)\eta_2]\cdot\zeta. \tag{36}$$

From (30) and (32), it is seen that as time gets large the filtered displacement r and the current perturbations η_1 and η_2 approach zero [2]. Thus, the rotor displacement x goes to zero. Furthermore, since η_1, η_2, and r go to zero, we can claim from (17) that $\tilde{c}$ also goes to zero as time gets large. That is, the estimated parameter vector $\hat{c}$ converges to the actual value c, as proven in Section IV below.

IV. Proof of Parameter Convergence

We now prove that the parameter estimate converges to the true value. This will be accomplished in two steps. First, we show that the parameter estimate converges to a constant vector and thus the parameter estimate error converges to a constant vector. Then, we prove that this parameter error vector is zero. To begin the proof, three theorems are presented as follows.

Theorem 1 *If a system is globally asymptotically stable and locally exponentially stable around its origin, then the system is globally exponentially stable around its origin.*

The proof of this theorem is omitted since it is quite straightforward following the definitions of asymptotical and exponential stabilities (global and/or local). Next, a theorem is cited from [4] and will not be proven.

Theorem 2 *For a nonlinear system, if its corresponding linearized system around its origin is globally exponentially stable, then the nonlinear system itself is locally exponentially stable around its origin.*

The last theorem is specific to the system considered and is obtained using Theorem 2.

Theorem 3 *Let* $\gamma = [r\ \eta_1\ \eta_2]^T$ *denote the state of the error dynamics consisting of (17), (28) and (29). Then the dynamic of the error signal* γ *is locally exponentially stable around its origin* $\gamma = 0$.

Now, we are ready to deduct the parameter convergence. Because of (30) and (32), we know from the LaSalle-Yoshizawa Theorem that $\lim_{t\to\infty} \gamma = 0$, i.e., the error signal γ is globally asymptotically stable at its origin. Combining this with Theorem 3, we know that $\gamma = 0$ is a globally exponentially stable equilibrium of the error dynamics, i.e., γ belongs to L_p for all $p \in [1, \infty)$ (cf. [5]) and thus $\frac{d}{dt}\{\hat{c}\}$ given by (36) belongs to L_1. It follows that the integral from $t = 0$ to $t = \infty$ of the parameter update law in (36) exists [3]. So the parameter estimate error $\tilde{c}$ approaches a constant vector $C = [C_1\ C_2\ C_3]^T$, as time gets large. We must now show that C is a zero vector.

Let $\tilde{c}(t) = C + \delta_c(t)$ and $\Delta t = \pi/2\Omega$, where $\delta_c(t) = [\delta_{c1}(t)\ \delta_{c2}(t)\ \delta_{c3}(t)]^T$. Note that $A(t)$ and $B(t)$ defined in (18) and (19) respectively are bounded, i.e., $|A(t)| \leq \bar{A}$ and $|B(t)| \leq \bar{B}$. Since $\delta_c(t)$, $r(t)$, $\eta_1(t)$, and $\eta_2(t)$ approach zero as time gets large, there exists a time t_L such that

$$|\delta_c(t)| \leq \frac{\Omega}{2(4+\pi)}\varepsilon, \tag{37}$$

$$|r(t)| \leq \frac{1}{4(2m + \Delta t \cdot k_v)}\varepsilon, \tag{38}$$

$$|\eta_1(t)| \leq \frac{1}{4\Delta t \cdot \bar{A}}\varepsilon, \tag{39}$$

$$|\eta_2(t)| \leq \frac{1}{4\Delta t \cdot \bar{B}}\varepsilon, \tag{40}$$

for $t \geq t_L$ and any arbitrarily small positive number ε. We now integrate the both sides of (17) over the time interval $[t,\, t+\Delta t]$ for $t \geq t_L$, as shown

$$\int_t^{t+\Delta t} m\dot{r}dt = \int_t^{t+\Delta t} -k_v r(t)dt + \int_t^{t+\Delta t} \tilde{c}^T(t)\zeta(t)dt + \int_t^{t+\Delta t} A(t)\eta_1(t)dt + \int_t^{t+\Delta t} B(t)\eta_2(t)dt. \tag{41}$$

Using the mean-value theorem for integrals, we rewrite (41) as

$$\begin{aligned} C^T \int_t^{t+\Delta t} \zeta(t)dt = &-\Delta t \cdot \delta_{c1}(\xi_1) - \delta_{c2}(\xi_2) \int_t^{t+\Delta t} \sin(\Omega t)dt \\ &-\delta_{c3}(\xi_3) \int_t^{t+\Delta t} \cos(\Omega t)dt + m[r(t+\Delta t) - r(t)] \\ &+k_v \Delta t \cdot r(\xi_4) - \Delta t \cdot A(\xi_5)\eta_1(\xi_5) \\ &-\Delta t \cdot B(\xi_6)\eta_2(\xi_6), \end{aligned} \tag{42}$$

where $\xi_i \in [t, t+\Delta t]$ $(i = 1, 2, \cdots, 6)$. From (37) through (40), and (42), we can obtain

$$\left| C^T \int_t^{t+\Delta t} \zeta(t)dt \right| \leq \varepsilon, \qquad (t \geq t_L). \tag{43}$$

Because the integral interval $\Delta t = \pi/2\Omega$ is a quarter of the period of the sinusoidal function representing the unbalance vibration, we can choose t $(t \geq t_L)$ such that

$$\left| C^T \int_t^{t+\Delta t} \zeta(t)dt \right| = |C_1| \cdot \frac{\pi}{2\Omega} + |C_2| \cdot \frac{1}{\Omega} + |C_3| \cdot \frac{1}{\Omega}. \tag{44}$$

We combine (43) and (44) to conclude that C is a zero vector. This ends the proof of the parameter convergence.

V. Simulation

To show the performance of the proposed control strategy, we present simulation results using the system parameters from an actual AMB system described in [9] (note, the value of δ was obtained from [15])

$$m = 7\mathbf{kg},\ g = 9.8\mathbf{m/s^2},$$

$$k = 6.93 \times 10^{-5}\mathbf{N \cdot m^2 \cdot A^{-2}},\ w = 0.7\mathbf{mm},$$

$$R = 14.7\mathbf{\Omega},\ L = 285\mathbf{mH},\ \delta = 1.233 \times 10^{-5}\mathbf{m}.$$

The initial conditions are $x(0) = 0.6\mathbf{mm}$, $\dot{x}(0) = 10\mathbf{mm/s}$, $I_1(0) = 0\mathbf{A}$, and $I_2(0) = 0\mathbf{A}$. The disturbance is shown in Figure 2 where a static load change of 50N is applied at $t = 15\mathbf{sec}$ and a sinusoidal disturbance $10\sin(10t + 0.8)$, i.e., $6.97\sin(10t) + 7.17\cos(10t)$, representing the effect of unbalance is applied at $t = 30\mathbf{sec}$. Initial value of $\hat{c}$ is $\hat{c} = [49\ 8\ 8]^T$ (Note, the notation here is not rigorous). The control gains and parameters are selected to be

$$\alpha = 20,\ k_v = 10,\ \sigma = 1 \times 10^4,$$

$$\Gamma_1 = 10,\ \Gamma_2 = 10, k_{c1} = k_{c2} = k_{c3} = 20000.$$

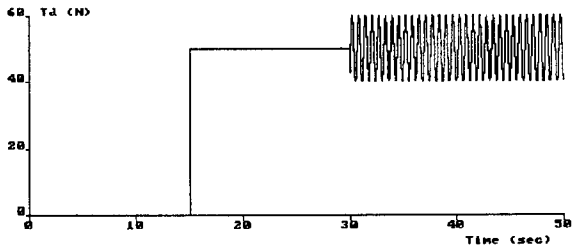


Figure 2. Disturbance in the vertical axis.

The dynamic behavior of the rotor position is shown in Figure 3. Note, when there is no disturbance the rotor is driven to the stator center from its initial position of $x(0) = 0.6\mathbf{mm}$ (due to the scale of the time axis, the initial position is not clear in Figure 3); when a step disturbance is applied the rotor is disturbed but returns to the stator center after a short time; when the unbalance is added the rotor is disturbed again and then goes to the stator center without any steady-state error. Note, the magnitudes and initial phase angle of the disturbances are unknown to the controller.

Figure 3. Rotor position under disturbance.

Figure 4 shows the control voltages V_1 and V_2 input to the top and bottom electromagnets, respectively.

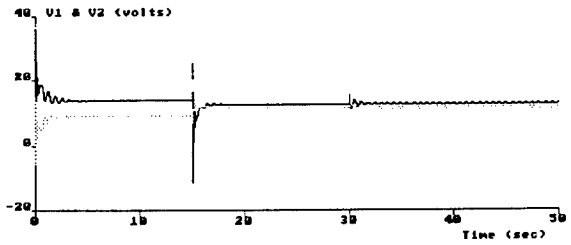


Figure 4. Control voltages.

Figure 5 through Figure 7 show the evolution of the parameter estimates. We see that the parameter estimate $\hat{c}$ does converge to the actual value $c = [50\ 6.97\ 7.17]^T$.

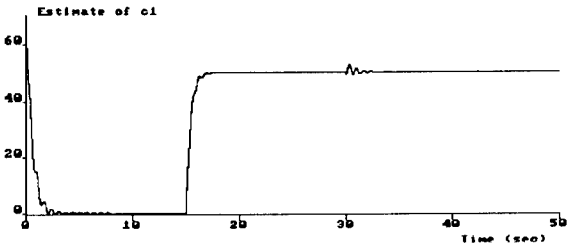


Figure 5. Estimate of c_1 (i.e., evolution of $\hat{c}_1$).

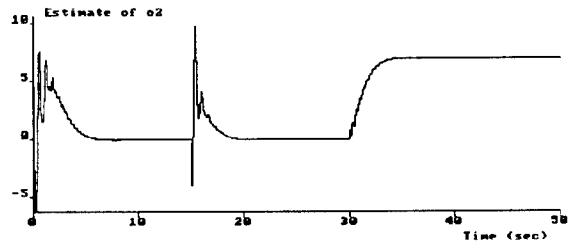


Figure 6. Estimate of c_2 (i.e., evolution of $\hat{c}_2$).

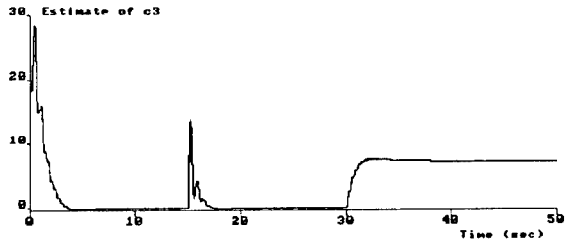


Figure 7. Estimate of c_3 (i.e., evolution of $\hat{c}_3$).

VI. Conclusion

Based on a backstepping technique, a voltage-level adaptive controller is designed for a single axis of an AMB system undergoing unknown static load changes and rotor imbalance. The disturbed rotor is controlled to rotate about its geometric axis. Estimates of the unknown rotor parameters converge to their true values. Simulation results verify the theoretical developments. Future work in this area is needed to extend the results to cases where the speed of the rotor is unknown and/or variable.

References

[1] Dawson, et al, "Nonlinear control of AMB systems: a backstepping approach," *IEEE Trans. Contr. Syst. Tech.*, Sept. 1996.

[2] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*, New York, NY: John Wiley & Sons, Inc., 1995.

[3] M. Krstić, "Invariant manifolds and asymptotic properties of adaptive nonlinear stabilizers," accepted to *IEEE Trans. Autom. Contr.*, Nov. 1995 (ACC'95 Best Student Paper).

[4] H. K. Khalil, *Nonlinear Systems*, New York, NY: Macmillan Publishing Company, 1992.

[5] M. Vidyasagar, *Nonlinear Systems Analysis*, Englewood Cliffs, NJ: Prentice Hall, Inc., 1993.

[6] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.

[7] H. Bleuler, et al, "Application of digital signal processors for industrial magnetic bearings," *IEEE Trans. Contr. Syst. Tech.*, vol. 2, no. 4, pp. 280-289, Dec. 1994.

[8] B. Shafai, S. Beale, P. LaRocca, and E. Cusson, "Magnetic bearing control systems and adaptive forced balancing," *IEEE Contr. Syst. Mag.*, vol. 14, no. 2, pp. 4-13, April 1994.

[9] F. Matsumura and T. Yoshmoto, "System modeling and control design of a horizontal-shaft magnetic bearing system," *IEEE Trans. Magnetics*, vol. MAG-22, no. 3, pp. 196-203, May 1986.

[10] K. Nonami and H. Yamaguchi, "Robust control of magnetic bearing systems by means of sliding mode control," *Proc. 3rd Int. Symp. Magnetic Bearings*, Alexandria, VA, pp. 537-546, July 1992.

[11] R. D. Smith and W. F. Weldon, "Nonlinear control of a rigid rotor magnetic bearing system: modeling and simulation with full state feedback," *IEEE Trans. Magnetics*, vol. 31, no. 2, pp. 973-980, March 1995.

[12] G. Schweitzer, Ed., *Proc. First Int. Symp. Magnetic Bearings*, ETH Zurich, Switzerland, June, 1988.

[13] T. Higuchi, Ed., *Proc. Second Int. Symp. Magnetic Bearings*, Tokyo, Japan, July 1990.

[14] P. E. Allaire, Ed., *Proc. Third Int. Symp. Magnetic Bearings*, Alexandria, VA, 1992.

[15] Y. Lu and J. Chen, "Design of a perturbation estimator using the theory of variable-structure systems and its application to magnetic levitation systems," *IEEE Trans. Industrial Electronics*, vol. 42, no. 3, pp. 281-289, June 1995.

[16] Y. Lu and J. Chen, "A self-organizing fuzzy sliding-mode controller design for a class of nonlinear servo systems," *IEEE Trans. Industrial Electronics*, vol. 41, no. 5, Oct. 1994.

[17] A. M. Mohamed and F. P. Emad, "Conical magnetic bearings with radial and thrust control," *IEEE Trans. Autom. Contr.*, vol. 37, no. 12, pp. 1859-1868, Dec. 1992.

[18] D. Vischer and H. Bleuler, "Self-sensing active magnetic levitation," *IEEE Trans. Magnetics*, vol. 29, no. 2, pp. 1276-1281, March 1993.

[19] D. Cho, et al, "Sliding mode and classical controllers in magnetic levitation systems," *IEEE Contr. Syst. Mag.*, vol. 13, no. 1, pp. 42-48, Feb. 1993.

[20] F. Matsumura, M. Fujita, and C. Oida, "Theory and experiment of magnetic bearing combining radial control and thrust control," *IEEE Trans Magnetics*, vol. MAG-23, no. 5, pp.2581-2583, Sep. 1987.

[21] M. Fujita, et al, "μ-Synthesis of an electromagnetic suspension system," *IEEE Trans. Autom. Contr.*, vol. 40, no. 3, pp. 530-536, March 1995.

[22] A. M. Mohamed and F. P. Emad, "Nonlinear oscillation in magnetic bearing systems," *IEEE Trans. Autom. Contr.*, vol. 38, no. 8, pp. 1242-1245, August 1993.

[23] M. Fujita, K. Hatake, and F. Matsumura, "Loop shaping based robust control of a magnetic bearing," *IEEE Contr. Syst. Mag.*, vol. 13, no. 4, pp. 57-65, August 1993.

[24] D. L. Trumper, "Nonlinear compensation technique for magnetic suspension systems," *NASA Workshop on Aerospace Application of Magnetic Suspension Technology*, Hampton, VA, pp. 378-388, Sept. 1990.

[25] A. M. Mohamed and I. Busch-Vishniac, "Imbalance compensation and automation balancing in magnetic bearing systems using the Q-parameterization theory," *IEEE Trans. Contr. Syst. Tech.*, vol. 3, no. 2, pp. 202-211, June 1995.