



AFRL-AFOSR-VA-TR-2021-0044

C1160: An Integrated Approach to Space Situational Awareness

Chakravorty, Suman
TEXAS ENGINEERING EXPERIMENT STATION COLLEGE STATION
1470 WILLIAM D FITCH PKY
COLLEGE STATION, TX,
US

06/02/2021
Final Technical Report

DISTRIBUTION A: Distribution approved for public release.
--

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 02-06-2021		2. REPORT TYPE Final		3. DATES COVERED (From - To) 15 Dec 2016 - 14 Dec 2020	
4. TITLE AND SUBTITLE C1160: An Integrated Approach to Space Situational Awareness				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-17-1-0068	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Suman Chakravorty				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) TEXAS ENGINEERING EXPERIMENT STATION COLLEGE STATION 1470 WILLIAM D FITCH PKY COLLEGE STATION, TX US				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AF Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/AFOSR RTA2	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-AFOSR-VA-TR-2021-0044	
12. DISTRIBUTION/AVAILABILITY STATEMENT A Distribution Unlimited: PB Public Release					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This project has led to fundamental advancements in the fields of Filtering/ Data Assimilation, Multi-Target Tracking and Data based Control. In Filtering, we have developed particle filters that are immune to the Curse of Dimensionality, called the Particle Gaussian Mixture Filters (PGM). We have also developed hybrid consensus and covariance intersection based distributed estimation algorithms that retain the robustness of CI while recovering the performance of consensus methods. In Multi-Target tracking (MTT), we have unified the hitherto thought to be different MTT techniques based on the classical Multi-Hypothesis tracking (MHT) and Random Finite Set (RFS) based methods. We have developed a highly efficient randomized technique for MTT, called the Randomized Finite Set Statistics (RFISST), that significantly outperforms classical MHT methods based on Munkres/ Murty's algorithms. The efficacy of the techniques has been shown for Space Situational Awareness (SSA) problems. The project has also contributed to the development of a Dynamic Data Driven Approach to Planning and Control of unknown systems/ Reinforcement Learning termed Decoupled data based Control (D2C) that offers a new and highly efficient paradigm for the feedback control synthesis of highly nonlinear and high dimensional systems, problems that were hitherto thought to be intractable. This technique breaks Bellman's infamous "Curse of Dimensionality" in nonlinear feedback control.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			ERIK BLASCH
U	U	U	UU	2	19b. TELEPHONE NUMBER (Include area code) 426-7311

AFOSR Annual/Final Report

Contract/Grant Title: An Integrated Approach to Space Situational Awareness

Contract/Grant #: FA9550-17-1-0068

Reporting Period: Final Report (Dec. 16, 2016- Dec. 15, 2020)

Author: Suman Chakravorty

Abstract

Main Accomplishments/Successes

A. Multi-Target tracking (MTT). We have shown the commonality between classical methods such as Multi-Hypothesis Tracking (MHT) and modern Random Finite Set (RFS) based methods for multi-target tracking. We have developed a randomized approach to MTT, termed the Randomized Finite Set Statistics (RFISST), that is capable of solving problems that were hitherto unsolvable using traditional methods like Munkres/ Murty. We have shown the application of these techniques to the Space Domain Awareness (SDA) problem.

B. Nonlinear Filtering. We have developed an unsupervised learning/ clustering based approach to nonlinear filtering problems, termed the Particle Gaussian Mixture (PGM) filters, that are immune from the Curse of Dimensionality in such problems. These filters have been shown to work on highly nonlinear and multi-modal problems, and their performance has been shown to be far superior traditional particle filtering techniques as well as Kalman filtering techniques.

C. Data based Control/ Reinforcement Learning. The project has also contributed to the development of a Dynamic Data Driven Approach to Planning and Control of unknown systems/ Reinforcement Learning termed D2C. These methods have been shown to be scalable to high dimensional problems, accurate and reliably converge to the correct solution, and are globally optimal in terms of their performance.

Research Contributions:

The research contributions include

1. W. Faber, S. Chakravorty and I. Hussein, "Tracking Space Collisional Cascading via Randomized Finite Set Statistics (FISST)", The Journal of the Astronautical Sciences, accepted
2. W. Faber, S. Chakravorty and I. Hussein, "A Belief Space Perspective on RFS based Multi-Target Tracking," IEEE Transactions on Aerospace and Electronic Systems, under review
3. D. Raihan and S. Chakravorty, "A Hybrid Particle-UKF for Space Object Tracking", J. Astronautical Sciences, 2017, DOI 10.1007/s40295-017-0114-8
4. D. Raihan and S. Chakravorty, "Particle Gaussian Mixture Filters I", Automatica, vol. 98, pp. 331-340, 2018
5. D. Raihan and S. Chakravorty, "Particle Gaussian Mixture Filters II", Automatica, vol. 98, pp. 341-348, 2018
6. Amirhossein Tamjidi, Suman Chakravorty and Dylan Shell, "Decentralized State Estimation via a Hybrid of Consensus and Covariance intersection", Autonomous Robots, May 2019, <https://doi.org/10.1007/s10514-019-09854-3>
7. A. Tamjidi, R. Otfadeh, S. Chakravorty, D. shell and D. Yu, "Unifying Consensus and Covariance Intersection for Distributed State Estimation over Unreliable Networks", IEEE Transactions on Robotics, to appear
8. K. Parunandi and S. Chakravorty, "T-PFC: A Trajectory-Optimized Feedback Control Approach", IEEE Robotics and Automation Letters, vol. 4, n. 4, pp. 3457-3463, Oct. 2019

9. R. Wang, R. Goyal, S. Chakravorty and R. E. Skelton, "Model and Data based Approaches to the Control of Tensegrity Robots", IEEE Robotics and Automation Letters, vol. 5, n. 3, pp. 3846-3853, 2019, DOI: 10.1109/LRA.2020.2979891
10. R. Wang, K. Parunandi, S. Chakravorty and D. Kalathil, "A Decoupled Data based Approach for Learning to Control Nonlinear Systems", IEEE Transactions on Automatic Control, to appear

**** Note:** just attach journal paper at end versus recopying the information in the text

Transitions: (required for final report)

1. Methods to industry

The RFISST, PGM and IRHC algorithms have been used for improved Space Catalog Processing, AFRL/ RV, under program manager Mr. Apoorva Bhopale. These algorithms were also used in an AF TENCAP program to provide orbit states using multiple sensor data types, Air Force TENCAP (ACC), under Program Manager Lt. Col. Paul Olah. They were integrated into the Automated Indications & Warnings for Orchestrated Command and Control (AIWOC) program, in conjunction with the Space Catalog Processing project, AFRL/RI, under program manager Ms. Carolyn Sheaff.

2. Students to employment (trained under grant)

Weston Faber, Senior Research Scientist, L3Harris

Dilshad Raihan, Research Engineer, Amazon Web Services (AWS)

3. Techniques to AFRL (correspondence with AFRL researchers)

The RFISST, PGM and IRHC techniques were transferred to AFRL researchers at AFRL/ RV and AFRL/ RI, please see above.

Collaborations:

1. AFRL

Scott Erwin, AFRL/ RV; Erik Blasch, AFRL/ RI; Muralidhar Rangaswamy, AFRL Sensors Directorate

2. DOD

N.A.

3. Others: e.g., NSF

NSF Computer Enabled Discovery, Science and Engineering (CD&SE)

4. Industry

N.A.

5. Student Internships

Utkarsh Mishra, Leolabs LLC

Mohamed NAVEED Gulohamed, GOOGLE X

Awards:

1. Best paper

The Hybrid Consensus and Covariance Intersection (HCCI) work was awarded the Best Paper award at the 2017 IEEE International Symposium on Multi Robot Systems (MRS) for “providing a robust and performant alternative for Distributed Estimation problems”

2. Recognition at school

3. External recognition

Publications:

1. Patents

N.A.

2. Journal

10

3. Conference

20

4. Posters

N.A.

Software:

Toolboxes available

N.A.

Invited Talks:

1. DOD meetings

S. Chakravorty, “Foundational Problems in Space Situational Awareness”, U. Colorado, Boulder, Feb. 2019

S. Chakravorty, “A Decoupling Principle in Stochastic Optimal Control and its Implications”, UC, Berkeley, March. 2019

S. Chakravorty, “A Decoupling Principle in Stochastic Optimal Control and its Implications”, USC, Los Angeles, April. 2019

S. Chakravorty, “A Decoupling Principle in Stochastic Optimal Control and its Implications”, UCLA, April. 2019

S. Chakravorty, “A Decoupling Principle in Stochastic Optimal Control and its Implications”, UC Irvine, April. 2019

S. Chakravorty, "A Decoupling Principle in Stochastic Optimal Control and its Implications", U. Michigan, Ann Arbor, Sep. 2019

S. Chakravorty, "Foundational Problems in Space Situational Awareness", U. Texas, Austin, Oct. 2019

S. Chakravorty, "A Decoupling Principle in Stochastic Optimal Control", Georgia Tech, Feb. 2020

S. Chakravorty, "A Decoupling Theory for Optimal Feedback Control", AFRL Space Vehicles Directorate, Aug. 2020

2. Professional Societies

3. Students

4. AFOSR PI Meeting

AFOSR DDIP PI meeting: 2016-2020

Community Efforts:

1. Data Collects

N.A.

2. Student events

N.A.

References:

Changes in Research Objectives: N.A.

Change in AFOSR Program Manager: N.A.

Extensions Granted or Milestones Slipped: N.A.

New discoveries, inventions, or patent disclosures: N.A.

Append 1 – 3 Published Papers to the PDF file before submitting

Application of a Randomized-Finite Set Statistics Technique (R-FISST) to Space Situational Awareness

W. R. Faber · U. R. Mishra
S. Chakravorty · I. I. Hussein

Received: date / Accepted: date

Abstract This paper presents a novel approach to keeping the Random Finite Set (RFS) based Bayesian recursions tractable. We propose a randomized scheme using a Markov Chain Monte Carlo (MCMC) based technique and Finite Set Statistics (FISST), termed Randomized FISST (R-FISST). This technique samples highly probable association hypotheses and uses them to approximate the posterior RFS based multi-object Probability Density Function (pdf). It samples hypotheses without enforcing a heuristic number of samples so the number of samples is able to adjust naturally to the ambiguity of the Data Association Problem (DAP). This provides the technique with a level of robustness to false associations. This is illustrated using two Space Situational Awareness (SSA) examples. In the first example we compare the R-FISST technique to the Global Nearest Neighbor (GNN) technique. GNN represents the extreme case where the tracking problem is kept tractable, but the approximation of the multi-object pdf lacks association diversity, which leads to divergence in cardinality. The second example shows the scalability of the R-FISST technique by demonstrating its performance on Collisional Cascading, i.e., a cascading space object fragmentation event.

Keywords Space Situational Awareness · Collisional Cascading · Space Traffic Management · Multiple Object Tracking · Markov Chain Monte Carlo · Finite Set Statistics

W. R. Faber
10807 New Allegiance Drive, Suite 401, Colorado Springs, CO 80921
Tel.: 1-719-284-6920x1195
E-mail: Weston.Faber@L3Harris.com

U. R. Mishra
301 RDMC Bldg, College Station, TX 77843
E-mail: utkarshranjanmishra@tamu.edu

S. Chakravorty
701 H.R. Bright Bldg, College Station, TX 77843-3141
E-mail: schakrav@tamu.edu

I. I. Hussein
10440 Little Patuxent Parkway, Suite 600, Columbia, MD 21044
Tel.: 1-410-715-0005
E-mail: ihussein@applieddefense.com

1 Nomenclature

t	Time
X	The set of objects
n	The cardinality of the object set
Z	The set of observations
m	The cardinality of observation set
$q^{(n)}$	A particular hypothesis consisting of n objects
$\omega_t^{q^{(n)}}$	Weight of a hypothesis containing n objects at time step t
ν	All possible permutations on the numbers 1 to n
ν_i	The i^{th} element of the permutation
$\sigma^{(n)}$	An n object data association hypothesis
σ_n^b	A birth hypothesis consisting of n birth objects
V	Sensor volume
P_d	Probability of detection
P_s	Probability of survival
λ_C, λ_B	Average clutter/ birth arrival rate

2 Introduction

In this paper, we present a randomized approach to approximate the full Bayesian recursions involved in solving the Random Finite Set (RFS) based approach to the problem of multi-object tracking and detection[15]. We begin by showing that the multi-object Bayesian recursions have a discrete hypothesis based structure analogous to Multiple Hypothesis Tracking (MHT)[5]. The ‘‘Hypothesis Space’’ perspective allows us to develop a randomized approach, termed Randomized Finite Set Statistics (R-FISST), where the highly probable association hypotheses are sampled using a Markov Chain Monte Carlo (MCMC) technique thereby allowing us to approximate the posterior multi-object pdf, and thus, allows for a computationally tractable implementation. The proposed technique is applied to space object tracking problems where the number of objects varies throughout the simulation. The performance is compared to the Global Nearest Neighbor (GNN) approach [16, 2], and is shown to be scalable using a Collisional Cascading event.

There are many approaches for solving the multi-object tracking problem. For example, the classical techniques such as Multiple Hypothesis Tracking (MHT) [19], Probabilistic Data Association (PDA)[2], and Joint Probabilistic Data Association (JPDA) [11][21]. More recently, a framework using Finite Set Statistics (FISST) has developed a statistically rigorous approach to the multi-object tracking problem. This has lead to the modern techniques such as Probabilistic Hypothesis Density (PHD) and Cardinalized-PHD (CPHD) [15, 24] as well as Generalized Labeled Multi-Bernoulli (GLMB) and its variations [22, 23, 12]. In this paper, a multi-object tracking technique called Randomized-Finite Set Statistics (R-FISST) [7, 9, 10] is used to track simulated space object fragmentation events. Fragmentations and space object collisional cascading cause data associations to become highly ambiguous. The number of resulting data association hypotheses

grows in a combinatorial fashion and is a function of the number of returns and the number of associable objects. It is further complicated when considering object birth or death. The exhaustive generation of all possible hypotheses can cause memory issues that lead to computational intractability. On the other hand, not generating a statistically relevant set of hypotheses can lead to inaccurate estimation and tracking divergence. The R-FISST tracking technique uses a Markov Chain Monte Carlo (MCMC) based sampling approach named Smart Sampling MCMC (SSMCMC) that allows one to sample the highly probable hypotheses without having to generate all possible hypotheses. SSMCMC keeps the DAP tractable while allowing for a diverse sample set that accurately represents the distribution of highly probable hypotheses.

We should mention here other MCMC based techniques that have been used in Multi-Target tracking, both in the RFS GLMB [12], and the MHT context [3, 18, 6, 17]. MCMC is an umbrella term for a very large class of methods, and in particular, the different versions vary in their choice of proposal distributions. We are different from the above references in that we use a different (non-Gibbs) proposal and in the fashion we handle target birth. Most importantly, the GLMB implementation is sensitive to the Bernoulli birth model while we can use a standard Poisson model that allows RFISST to be measurement driven, and birth objects based on measurements, similar to MHT. This is borne out by our empirical results where we are able to pick up a large number of new targets, such as a debris field in SSA, while limiting ourselves to a Poisson birth model.

The contribution of this paper is to illustrate the scalability of the R-FISST technique using simulated space object fragmentation events, including a large-scale collisional cascading event that results in the generation of over a hundred new objects in geostationary earth orbit (GEO). Results show that the technique keeps the DAP tractable and develops accurate estimates of the number of objects and their states.

The contributions of this paper are as follows:

- We propose a randomized MCMC based hypothesis generation technique, called R-FISST, to tractably approximate the FISST multi-target recursions.
- We apply our method to the SSA problem and perform a detailed comparison of our technique with GNN techniques and also touch upon the design parameters tuning.
- Illustrate the scalability of our method by tackling a realistic Collisional Cascading scenario.

2.1 Outline of the Paper

The rest of the paper is organized as follows. Sections 3 and 4 are presented here as abbreviation of [5] for the sake of completeness. Readers are directed to [5] for a more thorough treatment. In Subsection 3.1, we introduce the FISST MT-pdf and the MT Markov transition and likelihood functions. In Subsection 3.2, we derive the multi-target Bayesian recursions in the absence of target birth, while in Section 3.3, we show the relationship between the RFS and Hypothesis Oriented Multi Hypotheses Tracking (HOMHT) based approaches to multi-target tracking in terms of the hypothesis update equations. In Section 4, we introduce

the MCMC based randomized hypothesis selection technique that results in the R-FISST algorithm and perform a comparison with a Munkres algorithm based multi target tracker. In Section 5, we show applications of the R-FISST technique to Space Situational Awareness multi target tracking scenarios.

3 A Belief Space Perspective of RFS Bayesian Multiple target Tracking

This section provides a brief utilitarian outline of the Bayesian MT-*pdf* recursions, while all the relevant details can be found in [5].

3.1 Preliminaries.

Let $x \in \mathbb{R}^n$ denote the kinematic state of a single target. Let $z \in \mathbb{R}^q$ denote a measurement of the target state.

Single Target Transition Density. Let the state transition density of a single target be given by $p(x'|x)$, which typically corresponds to a noisy dynamics model: $x' = f(x, w)$, where w is a random white process noise term that encapsulates the model uncertainty.

Single Target Measurement Likelihood. It is defined as the single target measurement likelihood as $p(z|x)$, which flows from the measurement model: $z = h(x) + v$, where v is a sensor noise corrupting the measurement of the target.

Clutter Model. In addition, in multi-target tracking, since the sensor is typically a radar or optical telescope, there is a need to account for false alarms or background noise/ clutter. The clutter model is assumed to be Poisson distributed, i.e., the number of false arrivals is Poisson distributed with an arrival intensity of λ_C : $P(n) = e^{-\lambda_C V} \frac{(\lambda_C V)^k}{k!}$, where V is the sensor volume, k is the number of measurements, and the location of the alarms is uniform within the volume.

Birth Model. Since new targets may be born, for instance, from a collisional cascading event, the birth model is also Poisson with an arrival intensity of λ_B .

Target Detection Process. It is assumed that any target in the field of view of the sensor is detected with a probability $P_d < 1$, independent of any other target.

Target Survival Process. Any target instance survives with a probability $P_s < 1$, independent of any other target.

Multi-Target Measurement Likelihood. Given a set of observation $Z = \{z_1, z_2, \dots, z_m\}$, let $\sigma^{(n)} = (\sigma_1^{(n)}, \sigma_2^{(n)}, \dots, \sigma_n^{(n)})$ denote a data association hypothesis for n targets at the state coordinates $X = \{x_1, x_2, \dots, x_n\}$, where $|X| = n$. The association hypothesis, $\sigma_i^{(n)} \in \{z_1, z_2, \dots, z_m, \emptyset\}$, associates the i^{th} target to one of m measurements, $z_{\sigma_i^{(n)}}$, or to nothing, $\emptyset$. The multi-target likelihood conditioning on all such possible data associations $\sigma^{(n)}$, and using the Law of Total Probability, is given by:

$$p(Z|X, n) = \sum_{\sigma^{(n)}} p(Z|\sigma^{(n)}, X, n) p(\sigma^{(n)}|X, n). \quad (1)$$

The term $p(\sigma^{(n)}|X, n)$ denotes the a priori probability of the data association $\sigma^{(n)}$, given that there are n targets, and that it assigns exactly k measurements to clutter that is Poisson distributed, is given by: $p(\sigma^{(n)}|X, n) = p(\sigma^{(n)}|n) = P_d^{(m-k)}(1 - P_d)^{n-(m-k)} \frac{e^{-\lambda V} (\lambda V)^k}{k!}$, where P_d is the probability of detection of a target. Further, the likelihood of the measurement Z , given the data association hypothesis $\sigma^{(n)}$, is: $p(Z|\sigma^{(n)}, X, n) = \frac{k!}{V^k} \prod_{i=1}^n p(z_{\sigma^{(n)}}|x_i)$, assuming that the clutter is uniformly distributed in the sensor volume V .

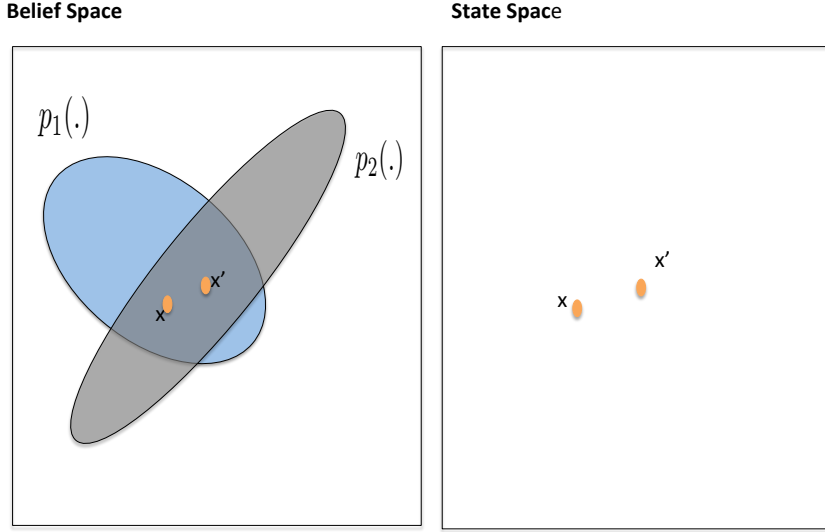


Fig. 1: The initial condition *pdf*. The *pdf* is symmetrized, i.e., written as $p(x, x') = p_1(x)p_2(x') + p_1(x')p_2(x)$, because the events E1: target 1 is at x and target 2 is at x' , and E2: target 1 is at x' and target 2 is at x , are indistinguishable. This phenomenon shows up on the right in a realization of the process in the state space where it is impossible to say if the sample x came from target/ *pdf* 1 and x' came from target/ *pdf* 2, or vice-versa. However, in the *pdf*/ belief space as shown on the left, the *pdf* of target 1 is **distinguishable** from that of target 2 (since the *pdfs* are distinct), i.e., there is no need for “labeling” them 1 and 2 to distinguish them. The symmetrized *pdf* representation above is simply a Mathematical representation of the belief space picture on the left.

3.2 Recursive Bayesian RFS based Multi-Target Tracking

This section shows the structure of the Bayesian multi-target recursions when the number of targets is fixed, the case with birth is relatively straightforward, albeit tedious, and left out (please see [5] for details).

Assume, initially, there exist n targets with distinguishable *pdfs*. The initial distribution can be represented as follows:

$$p_0(X) = \sum_{\nu} \prod_{i=1}^n p_0^i(x_{\nu_i}), \quad (2)$$

where ν is some permutation of the numbers $\{1, \dots, n\}$, the multi-target *pdf* has independent component target *pdfs* with $p_0^i(\cdot)$ denoting the “distinguishable” *pdf* of the i^{th} target, where the adjective distinguishable implies that $p_0^i(\cdot) \neq p_0^j(\cdot)$, for any $i \neq j$. An illustration of the concept of this “symmetrized *pdf*” (note the sum over permutations) is given in fig. 1.

Next, the multi-target *pdf* initial condition is recursively propagated and updated. This requires the notion of a “hypothesis” which can be defined constructively as follows (see Remark 1 below for a summary). Suppose the multi-target *pdf* at time $t-1$ can be represented by,

$$p_{t-1}(X, n) = \sum_{q^{(n)}} p_{t-1}^{q^{(n)}}(X) \omega_{t-1}^{q^{(n)}}, \quad (3)$$

where the sum is taken over all possible parent hypotheses $q^{(n)}$ containing n targets and $\omega_{t-1}^{q^{(n)}}$ is the corresponding weight. The first factor of Eq. (3) corresponds to the underlying multi-target state given the particular hypothesis $q^{(n)}$, and is expressed as: $p_{t-1}^{q^{(n)}}(X) = \sum_{\nu} \prod_i p_{t-1}^{q^{(n)}, i}(x_{\nu_i})$. Note that at $t=0$, there is a single hypothesis corresponding to the initial condition (2).

The predicted *pdf* at time t is given by:

$$p_t^-(X, n) = \sum_{q^{(n)}} p_t^{q^{(n)}, -}(X) \omega_{q^{(n)}}, \quad (4)$$

where $p_t^{q^{(n)}, -}(X) = \sum_{\nu} \prod_i p_t^{q^{(n)}, i, -}(x_{\nu_i})$, $p_t^{q^{(n)}, i, -}(x) = \int p(x|x') p_{t-1}^{q^{(n)}, i}(x') dx'$.

Proposition 1 *Given a measurement Z_t , the updated multi-target *pdf* is given by:*

$$p_t(X, n|Z_t) = \sum_{q^{(n)}} \sum_{\sigma^{(n)}} p_t^{q^{(n)}, \sigma^{(n)}}(X|Z_t) \omega_t^{q^{(n)}, \sigma^{(n)}}, \quad (5)$$

where

$$\omega_t^{q^{(n)}, \sigma^{(n)}} = \frac{\omega_{t-1}^{q^{(n)}} p(\sigma^{(n)}|n) l_{q^{(n)}, \sigma^{(n)}}}{\sum_q \sum_{r^{(q)}} \sum_{\nu^{(q)}} \omega_{r^{(q)}} p(\nu^{(q)}|q) l_{r^{(q)}, \nu^{(q)}}}, \quad (6)$$

$$l_{q^{(n)}, \sigma^{(n)}} = \frac{k!}{V^k} \prod_i \int p(z_{\sigma_i^{(n)}}^t | x) p_t^{q^{(n)}, i, -}(x) dx, \quad (7)$$

$$p_t^{q^{(n)}, \sigma^{(n)}}(X|Z_t) \equiv \sum_{\nu} \prod_i p_t^{q^{(n)}, \sigma^{(n)}, i}(x_{\nu_i} | z_{\sigma_i^{(n)}}^t), \quad (8)$$

$$p_t^{q^{(n)}, \sigma^{(n)}, i}(x | z_{\sigma_i^{(n)}}^t) = \frac{p(z_{\sigma_i^{(n)}}^t | x) p_t^{q^{(n)}, i, -}(x)}{\int p(z_{\sigma_i^{(n)}}^t | x') p_t^{q^{(n)}, i, -}(x') dx'} \quad (9)$$

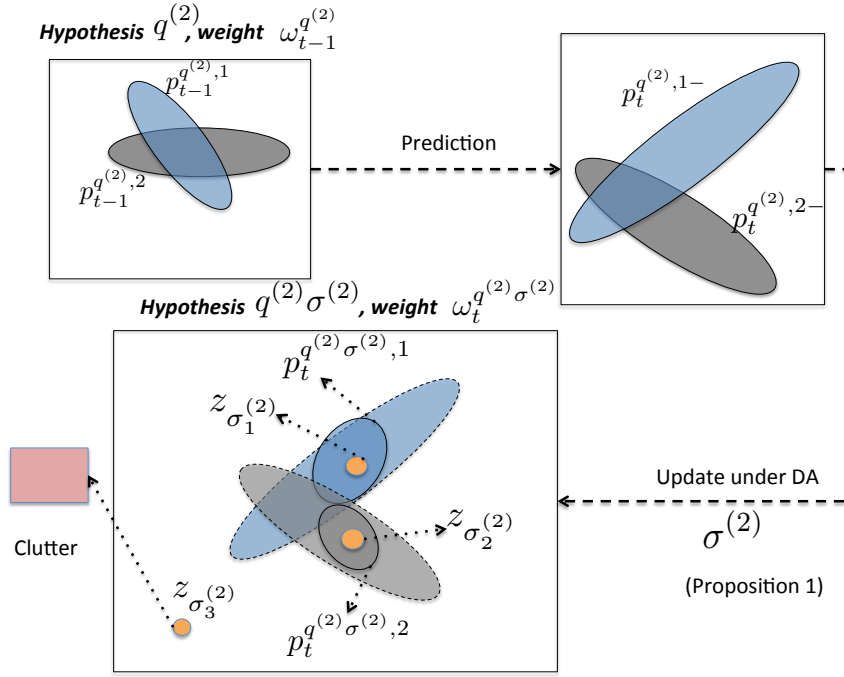


Fig. 2: An Illustration showing Bayesian prediction and update under FISST. The component *pdfs* 1 and 2 at time $t - 1$ under hypothesis $q^{(2)}$ are predicted to obtain the predicted prior at time t . Given the data association $\sigma^{(2)}$, the *pdfs* 1 and 2 are updated according to the measurements $z_{\sigma_1}^{(2)}$ and $z_{\sigma_2}^{(2)}$ respectively while $z_{\sigma_3}^{(2)}$ is associated to clutter. The complex structures of the FISST *pdfs* (in Eq. 9) is again due to the symmetrization necessary to account for the *indistinguishability of target states*. Moreover, the **data associations are to *pdfs* which are distinguishable and not to the “labels” 1,2; the labels are purely incidental and can be swapped without changing the MT-*pdf*.**

An illustration of the result above is shown in fig. 2.

Remark 1 Using the result above, and the initial condition in (1), it follows that a hypothesis $q^{(n)}$ at any time t , using MHT parlance, consists of compatible “tracks” of the initial set of targets, where a track is simply some initial target *pdf*, updated with a unique sequence of observations till time t , including the empty observation $\emptyset$. Two tracks are compatible if the same (non-empty) observation, is not assigned to two different targets. Hypotheses are hybrid entities with a discrete weight recursively calculated using (6), and a continuous underlying *pdf* given by (9).

3.3 Relationship Between Classical (HOMHT) and RFS based Multi-target Tracking Techniques

In this subsection, the RFS hypotheses update equations from Section 3.2 are simplified and related to the MHT (specifically Hypothesis Oriented Multiple Hy-

potheses Tracking (HOMHT)) hypotheses update equations in order to draw conclusive evidence about the relationship between the methods.

3.3.1 Relationship when the Number of Targets is Fixed

Consider a prior n -target hypothesis $q^{(n)}$, and assume no birth or death of targets, and the standard measurement model including missed detections and Poisson false alarms. Further, let the predicted densities of the targets underlying $q^{(n)}$ be given by $p^{q^{(n)},i-}(\cdot)$. The MHT posterior hypothesis probability, assuming that the data association $\sigma^{(n)}$ assigns exactly k measurements to clutter, is given by [19, pg. 847 Eq. 8]:

$$\omega_{MHT}^{q^{(n)},\sigma^{(n)}} = \eta p_d^{m-k} (1 - p_d)^{n-(m-k)} \lambda^k \prod_i \int p(z_{\sigma_i^{(n)}} | x_i) p^{q^{(n)},i-}(x_i) dx_i \omega_{q^{(n)}}. \quad (10)$$

This expression for the MHT posterior probability is exactly the same as the updated weight $\omega^{q^{(n)},\sigma^{(n)}}$ presented in Proposition 1 in Eq. 6. Moreover, the updated pdf s of the targets in MHT is identical to the FISST continuous update Eq. 9.

The above assertion becomes clear by noting that the factor $\int p(z_{\sigma_i^{(n)}} | x) p^{q^{(n)},i-}(x) dx$, under a Gaussian approximation, reduces to substituting $z_{\sigma_i^{(n)}}$ into the Gaussian pdf of the measurement innovation, $\mathcal{N}(\zeta - H\hat{x}^{q^{(n)},i-}; 0, HP^{q^{(n)},i-}H' + R)$, where the measurement equation has the linear form $z = Hx + v$, the measurement noise v has covariance R , and the i^{th} predicted pdf under hypothesis $q^{(n)}$, $p^{q^{(n)},i-}$, is assumed to be Gaussian with mean and covariance $\hat{x}^{q^{(n)},i-}$ and $P^{q^{(n)},i-}$: the case considered in Reid's paper. The fact that the target pdf update in MHT is identical to the FISST update Eq. 9, follows from noting that Eq. 9 reduces to the Kalman filter if we assume that the predicted pdf s are Gaussian. Thus, the FISST update can also be construed as a generalization of Reid's update to cases when the linear Gaussian approximation does not hold.

3.3.2 Relationship between HOMHT and FISST with Target Birth

Next is the case of a target birth, details can be found in [5]. Additionally, assume that $z = x + v$, i.e., a noisy full state measurement. Suppose now that underlying an HOMHT/ RFS hypothesis, there were r existing targets in the prior hypothesis $q^{(r)}$, $n - r$ births were hypothesized under the birth hypothesis σ_{n-r}^b , and exactly k measurements were associated to clutter under the data association $\sigma_a^{(n)}$. Note that in birth hypothesis σ_{n-r}^b , superscript b is not a number but simply a symbol to denote that the hypothesis is a birth hypothesis while the subscript $n - r$ denotes the number of new births. It can be seen from [19, pg. 848 Eq. 16] that, the HOMHT posterior hypothesis probability is,

$$\omega_{HOMHT}^{q^{(r)},\sigma_{n-r}^b,\sigma_a^{(n)}} = \eta (1 - p_d)^{n-(m-k)} p_d^{(m-k)} \left(\prod_{i=1}^r \int p(z_{\sigma_{a,i}^{(n)}} | x) p^{q^{(r)},i-}(x) dx \right) \left(\frac{\lambda_B}{p_d} \right)^{n-r} (\lambda_C)^k \omega_{q^{(r)}}, \quad (11)$$

where it should be noted that *under the HOMHT update, all $n - r$ births are detected, i.e., there is no provision in Reid's HOMHT update for detecting a subset s of the births ($s < n - r$)*. In general, this is true of MHT techniques in practice as well where new tracks are originated from every measurement [4]. In general, the RFS approach can have undetected births. Suppose now that in the RFS update, $s = n - r$, corresponding to the hypothesis that all hypothesized new births are detected. Then

$$\omega_{RFS}^{q^{(r)}\sigma_{n-r}^b\sigma_a^{(n)}} = \eta(1 - p_d)^{n-(m-k)} p_d^{(m-k)} \left(\prod_{i=1}^r \int p(z_{\sigma_{a,i}^{(n)}} | x) p^{q^{(r)}, i-}(x) dx \right) (\lambda_B)^{n-r} (\lambda_C)^k \omega_{q^{(r)}}. \quad (12)$$

Remark 2 Hypothesis Management. Note that the calculation of hypotheses weight in Eq. 11 and Eq. 12 are equivalent in all respects other than undetected new births and the $\frac{1}{p_d^{n-r}}$ factor in the MHT. However, it can be shown that the undetected birth case can be neglected in the RFS recursions without affecting the answer if the tracker is interested in only keeping a finite number of hypotheses, in the sense that there is always a hypothesis with “all detected births” that has higher weight than a hypothesis with “undetected births” (see [5]). Hence, given that the hypothesis space grows combinatorially fast, and we can only keep track of a finite number of hypothesis, it is only natural that we concentrate on the higher likely hypotheses. Thus, the most important aspect for any practical MTT technique, be it RFS based or MHT, is the tractable management of the combinatorially growing hypothesis space. In the following section, a computationally viable implementation of MTT recursion algorithm with good localization and cardinality tracking capability is presented.

4 Smart Sampling Markov Chain Monte Carlo

In this section, we describe a randomized approximation to the full FISST recursions called the R-FISST. A randomized hypothesis generation technique is used within R-FISST called the Smart Sampling Markov Chain Monte Carlo (SSMCMC) which can be used for hypothesis generation. In scenarios where generating all possible association hypotheses is computationally intractable, SSMCMC determines a set of hypotheses that can be used to approximate the multi-object posterior. It does this by creating a Markov chain using the particular association hypotheses as states of the chain. It then uses a biased random walk to explore the states of the chain. While exploring the chain, the technique uses an MCMC decision making criterion that favors the states with higher posterior probability and by doing so generates samples from the multi-object posterior distribution. The walk is performed until the assumed burn in period is reached. After the burn in period, the stationary distribution of the chain is sampled, and the resulting states are taken as the highly probable hypotheses. The algorithm for these steps is presented in Algorithm 1.

In the following section, the process used to perform the SSMCMC sampling is revealed. This is done by using a small SSA example. The first step in the SSMCMC process is to create the data association matrix D . The following construction is performed in a manner such that it is compatible with the Munkres

Algorithm 1 SSMCMC Hypothesis Generation

```

1: Create data association matrix  $D$ 
2: Generate child hypothesis  $\Sigma_k$ ,
3: set  $k = 0$ .
4: Generate  $\Sigma_{k+1} = \pi(\Sigma_k)$  where  $\pi(\cdot)$  is the proposal distribution
5: MCMC Criterion: If  $\omega_{\Sigma_{k+1}} > \omega_{\Sigma_k}$ 
6:    $\Sigma_k := \Sigma_{k+1}$ ;
7:   else  $\Sigma_k := \Sigma_{k+1}$  with probability proportional to  $\frac{\omega_{\Sigma_{k+1}}}{\omega_{\Sigma_k}}$ ;
8:   end
9: set  $k = k + 1$ .
10: repeat steps 4-9 until assumed burn in is reached
11: after burn in keep repeated  $\Sigma_k$ 

```

algorithm [16], i.e., it can be converted to a score matrix and the global optimal assignment can be determined. When constructed in this manner D will have dimensions $n + m \times 2n + 2m$, where n is the number of observable objects and m is the number of observations. It will contain elements that represent all relevant association types including those that correspond to object birth, object death, object missed detection, observation to object, and false alarm associations. Construction of the data association matrix is most easily described using eight sub matrices.

$$D = \begin{bmatrix} D_1 & D_2 & D_3 & D_4 \\ D_5 & D_6 & D_7 & D_8 \end{bmatrix}. \quad (13)$$

The first sub matrix D_1 is an $m \times n$ matrix that contains the information pertaining to observation-to-object association. Assuming the object has a probability of survival P_s and probability of detection P_d , then D_1 can be expressed as follows.

$$D_1 = [d_{ij} = \frac{P_s P_d p(z_i | x_j)}{(1 - P_d)}] \quad (14)$$

where d_{ij} are elements of D_1 with $i = 1, \dots, m$ and $j = 1, \dots, n$ being the row and column indexes respectively.

Remark 3 In Eq. 14, the denominator $(1 - P_d)$ is not a common factor of the individual observation-to-object association probability. However, it is a necessary factor due to this particular construction of the data association.

Sub matrix D_2 has dimensions $m \times m$ and contains the information for association to false alarm. In order to avoid breaking row and column rules D_2 is a diagonal matrix. Assuming that false alarm arrival is Poisson with average arrival rate λ_C and arbitrary spatial distribution according to $C(z_i)$ then D_2 is constructed as follows,

$$D_2 = \text{diag}(\lambda_C C(z_i)). \quad (15)$$

Where $i = 1, \dots, m$. D_3 is also an diagonal matrix where each element of the diagonal corresponds to a different birth association. Assuming birth arrival is Poisson with average arrival rate λ_B and limiting the number of births to the number of measurement returns the sub matrix D_3 is an $m \times m$ matrix and can be expressed as,

$$D_3 = \text{diag}(\lambda_B P_d p_B(z_i | x_i)). \quad (16)$$

Limiting the number of births to the number of observations makes $i = 1, \dots, m$. Also, assuming the birth pdf spatial distribution is uniform in the FOV allows for the convenient simplification $p_B(z_i|x_i) = \frac{1}{V}$. Another common assumption is to assume a spatial distribution that is normal centered around the observation. Matrices D_4 , D_6 , and D_7 are all zero matrices used as place holders in order to prevent breaking row and column rules. In large scenarios, these sub matrices cause the data association matrix to be sparse, but issues can be avoided by using sparse matrix tools that reduce extraneous computation and memory usage. D_5 is an $n \times n$ diagonal matrix that accounts for the probability of associating the objects to death. The expression for D_5 can be written as,

$$D_5 = \text{diag}((1 - P_s)). \quad (17)$$

In which, i takes values $1, \dots, n$. The probability of survival p_s can be state dependent $p_s(x_i)$ but is abbreviated for clarity. Lastly, D_8 is an $n \times n$ diagonal matrix that accounts for missed detections and can be expressed by,

$$D_8 = \text{diag}((1 - P_d)). \quad (18)$$

Where i goes from $1, \dots, n$.

This construction of the D matrix allows for the simple conversion into a score matrix that can be used in Munkres algorithm. It is also set up so that any valid selection of $n + m$ elements creates a valid grand child hypothesis that corresponds to a specific combination of parent hypothesis q , data association hypothesis θ , and birth and death hypothesis σ . The product of these elements then becomes the numerator of the discrete weight of the posterior pdf as in the previous section. Averaging over all possible θ and σ hypotheses provides the posterior weight for the given parent hypothesis q . This ensures that when SSMCMC is performed samples are taken from the posterior distribution and the corresponding set of highly probable hypotheses q^* accurately quantifies the grandchild hypothesis population. Using Eqs. 13-18 and assuming that the prior parent hypothesis q contains one object $X = x_1$ the data association matrix for an example problem that includes the one prior object and two observation returns,

$$D = \begin{bmatrix} \frac{P_d P_s p(z_1|x_1)}{1-P_d} & \lambda_C C(z_1) & 0 & P_d \lambda_B B(x) & 0 & 0 \\ \frac{P_d P_s p(z_2|x_1)}{1-P_d} & 0 & \lambda_C C(z_2) & 0 & P_d \lambda_B B(x) & 0 \\ (1 - P_s) & 0 & 0 & 0 & 0 & (1 - P_d) \end{bmatrix} \quad (19)$$

Creating the data association matrix has no added computational cost because each element of the matrix is necessary in calculating hypothesis probability and is used in every hypothesis generation technique. Also, the dimensions of the data association matrix are much smaller than those of the hypothesis matrix. This makes it much more practical to explore using MCMC.

After creating the data association one can begin the MCMC procedure. In the MCMC procedure one iterates through different values of a $2n + 2m \times n + m$ permutation matrix M . Each different M corresponds to a different grandchild hypothesis. To start the derivation, one must choose an initial $M = M_0$. This

can be done randomly or for faster convergence can be initialized with the global maximum likelihood. An example of such a matrix is as follows,

$$M_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

To be a valid permutation matrix that represents a grandchild hypothesis an M matrix must have the following properties.

- matrix elements must be either zero or one:

$$m_{ij} \in \{0, 1\}.$$

- The sum of any particular column j must be unity:

$$\sum_{i=1}^{n+m} m_{ij} = 1.$$

- The sum of any particular row i must be less than or equal to one:

$$\sum_{j=1}^{2n+2m} m_{ij} \leq 1.$$

- It must not contain any non-zero elements corresponding to zeros in the data association matrix:

$$m_{ij} \neq 1 \text{ if } D_{ji} = 0.$$

Equation 20 has all properties for a valid permutation matrix and pertains to a particular grandchild hypothesis in which, one observation is associated to a previously known object and another observation is associated to a predicted birth hypothesis.

The next step is to begin the random walk by setting the initial permutation matrix to the current permutation matrix M_k and determining the probability of the corresponding hypothesis. This can be done by taking the product of M_k and D creating a matrix where the nonzero elements correspond to the factors within the posterior weight. The weight is then the product of the nonzero elements. An example is shown using Eq. 20,

$$M_0 D = M_k D \quad (21)$$

$$= \begin{bmatrix} \frac{p_d P_s p(z_1|x_1)}{1-p_d} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_d \lambda_B B(x) & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-p_d) \end{bmatrix}. \quad (22)$$

With some abuse of notation, $M_k D > 0$ signifies the non zero elements of the $M_k D$ matrix. The nonzero elements are,

$$M_k D > 0 \implies \left[\frac{P_d P_s p(z_1|x_1)}{1-p_d} \quad P_d \lambda_B B(x) \quad (1-P_d) \right]. \quad (23)$$

Taking the product of the nonzero elements we get the discrete weight,

$$\omega_k = P_d^2 P_s p(z_1|x_1) \lambda_B B(x). \quad (24)$$

By including factors, like those from Poisson process models, that are missing from the data association matrix the numerator of the posterior weight can be recovered exactly to ensure the hypotheses are being sampled according to the posterior distribution.

After the current weight is determined, the next step is to determine a proposed state M_{k+1} to walk to. Typical randomized hypothesis generation techniques determine the proposed hypothesis by switching a single data association within the current hypothesis according to uniform proposal distribution. SSMCMC can perform either the uniform proposal approach or use a biased proposal distribution to generate a smart data association switch. This is extremely important when handling large matrices because the probability of uniformly sampling a highly likely hypothesis is vanishingly small. The smart switch is done by randomly sampling a row i and sampling the association j to switch to according the distribution of the row i . For the example shown, randomly sample a row i of D then sample j from i according to the Probability Mass Function (PMF) for the row i . After the switch is chosen, make the corresponding switch in M_k to create M_{k+1} . Assume the proposed switch was $i = 2$ and $j = 3$. The proposed hypothesis M_{k+1} is then,

$$M_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (25)$$

Next determine the weight of the proposed hypothesis $\bar{\omega}_{k+1}$ and compare it to the weight of current hypothesis $\bar{\omega}_k$ using an MCMC criterion that favors the more probable hypothesis. The expression for the MCMC criterion when sampling according to a uniform proposal distribution is,

$$M_k = \begin{cases} M_{k+1}, & \text{if } \frac{\bar{\omega}_{k+1}}{\bar{\omega}_k} \geq U([0, 1]) \\ M_k, & \text{otherwise} \end{cases}. \quad (26)$$

In words, if the probability of the proposed hypothesis divided by the probability of the current hypothesis is greater than a uniformly generated random number between zero and one then keep the proposed hypothesis and use it as the new current hypothesis. If not, keep the current hypothesis. It is important to note that if one choses to use the biased proposal distribution when sampling, the MCMC criterion must be modified to reflect a Metropolis-Hastings approach. In our case we have not included a Hastings symmetrization term but still the filter works well as can be seen from the results. Further investigation however, is needed into the calculation of the Hastings term in this formulation.

Continue the random walk until the assumed burn in period is over. After burn in, continue walking and take the set of sampled hypotheses as the set of highly probable hypotheses q^* . The data association matrix can be modified to include any type of association including but not limited to multiple maneuver, multiple

model, and spawn type associations. It is also compatible with different Markov motion models and different measurement models predicated on the correct formulation of the multi-target filter. Note that the ranked highly likely hypotheses can also be generated using deterministic schemes like Murthy's m-best but they are computationally very expensive as shown in fig. 3.

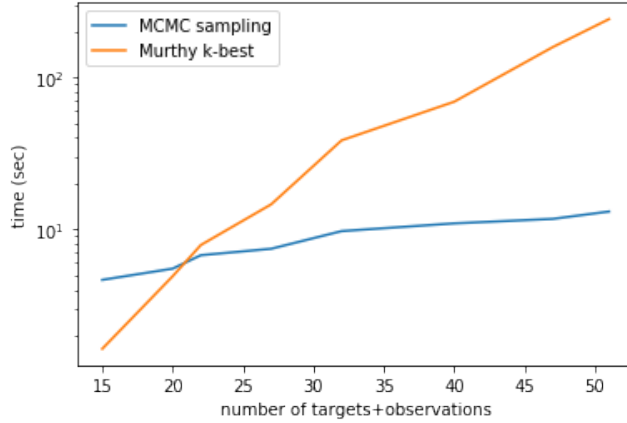


Fig. 3: Time taken by murthy's m best ($m = 10$) and SSMCMC sampling (10,000 burn-in and sampling steps each) vs size of the Data Association Matrix

4.1 Parameter-tuning and Comparision

Next we look at an edge case that can occur when tracking RSO with an insufficient number of data association hypotheses. To achieve this goal space object tracking scenarios are simulated. The MHT with Munkres algorithm is implemented to simulate a tracking algorithm that does not retain a sufficient number of hypotheses. The Munkres algorithm solves the DAP for the maximum likely data association hypothesis resulting from MHT recursion. This single hypothesis is used in a Global Nearest Neighbour(GNN) fashion throughout the simulation. This is often done in practice as a work around for situations where the number of possible hypotheses is too large to generate exhaustively. RSOs were randomly initialized in either Lower Earth Orbit (LEO) or Medium Earth Orbit (MEO).

The simulation was run long enough such that each object passed through the field of view a minimum of one time. The state vector for each object contained the objects position and velocity in the Cartesian frame. The dynamics model used was Two-body problem with no process noise. Clutter was added and assumed to be Poisson distributed with an average arrival rate $\lambda_C = 10$. The R-FISST technique was applied and given the initial means and covariances for all of the objects. No field of view restrictions were imposed i.e. it was assumed all the targets are visible at all the time. The sensor model measured full states with a $R = \text{diag}[(0.1 \text{ km})^2, (0.1 \text{ km})^2, (0.1 \text{ km})^2, (0.001 \text{ km/s})^2, (0.001 \text{ km/s})^2, (0.001 \text{ km/s})^2]$

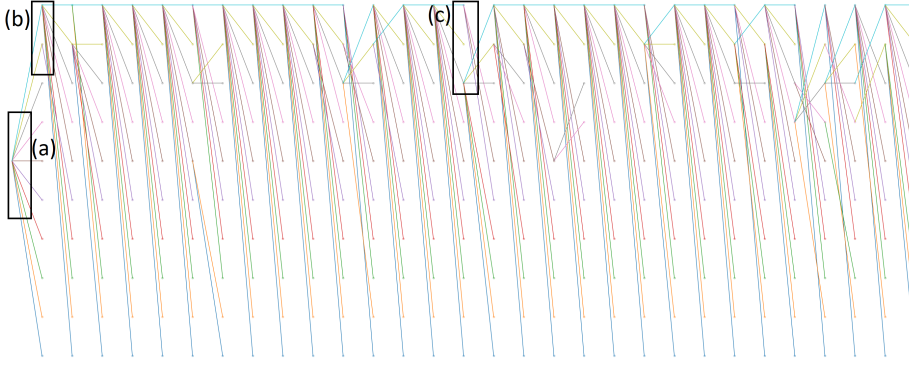


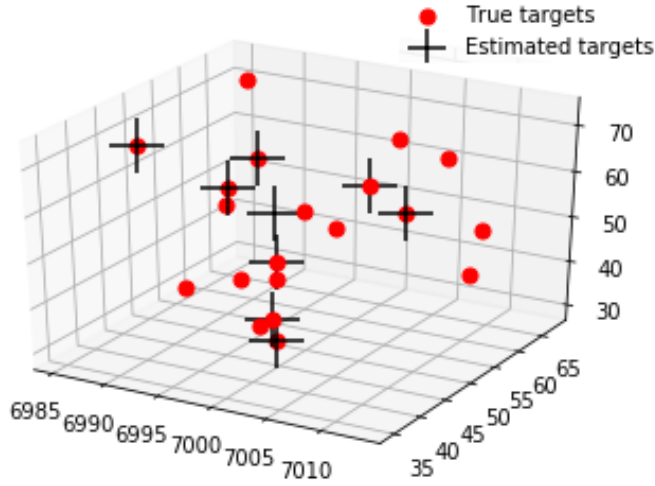
Fig. 4: Hypotheses tree containing top ten hypotheses at each time step. Insert (a) shows the root hypotheses/initial target states at t_0 which gives birth to 10 hypotheses at t_1 . Insert (b) shows that the top 10 hypotheses at t_2 are children of only the top 2 hypotheses at t_1 . Insert (c) shows that a lower ranked hypotheses at the current time step (third most likely in this case) can give birth to the top hypotheses at next time step.

The probability of detection inside the FOV was, $P_d = 0.9$. Measurements were taken at a standard interval that resulted in a total number of 35 scans throughout the simulation. Underlying continuous states were updated using an Extended Kalman Filter (EKF). From figure 6 one can see that Munkres implementation fails to initialize new targets. This is due to the fact that the implementation did not retain a sufficient number of hypotheses throughout the simulation.

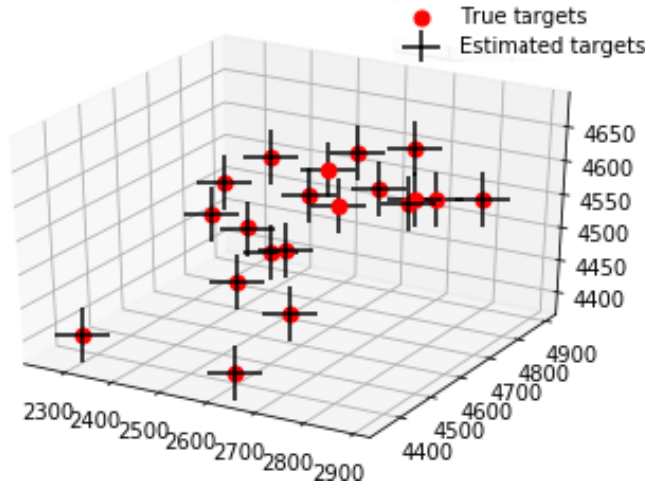
The same simulation was performed using the R-FISST technique. However, to increase the difficulty, the R-FISST technique was given an initial hypothesis that contained a joint point pdf with the initial means and covariances for some but not all of the RSOs. The MCMC in the R-FISST technique was assumed to converge after a maximum of ten thousand steps and one thousand steps were used to generate the retained likely hypotheses after convergence. Figure 5 shows the object position estimates before (figure 5a) and after (figure 5b) the objects are detected. From the illustration one is able to see that the R-FISST technique keeps the problem computationally tractable while retaining a sufficient number of hypotheses to track all new objects.

Next, three simulations were set up with varying number of true fragments (15, 20, 25) to see how well does the filter estimate cardinality. The filter was initialized with just 9 fragments and allowed to process sequential measurements. It is seen in fig. 6 that given sufficient time filter picks up on all the targets. This means that the filter is able to initialize new targets from measurement data.

Finally, four simulations were set up with differing values of λ_B where λ_B is the scaling term that goes into D_3 matrix in Eq. 16. By tuning the λ_B/λ_C while keeping λ_C constant in fig. 7, we observed that increasing the value of λ_B leads to quicker growth in the number of initialized targets. This is in accordance with what we should be expecting as λ_B is essentially a tuning knob that modulates the probability of arrival of new targets.



(a) True targets and R-FISST generated object positions illustrated before the objects are detected



(b) True targets and R-FISST generated object positions illustrated after the objects are detected

Fig. 5: All objects are captured and tracked when using the R-FISST technique

Thus, an important design choice is the λ_B/λ_C tuning. Figure 7 shows the effect of tuning this ratio on cardinality of the top hypothesis. Either the λ_B and λ_C can be set as constants or they can be linked to values in D_1 matrix. The latter approach helps in scaling the values of terms in D_1 vs values in D_2 and D_3 matrices appropriately. In addition to this P_s or probability of survival term also plays a role in retaining new births, removing wrong initializations and thereby correcting the cardinality of hypotheses. More investigation in terms of a multidimensional optimization over these design variables is being pursued.

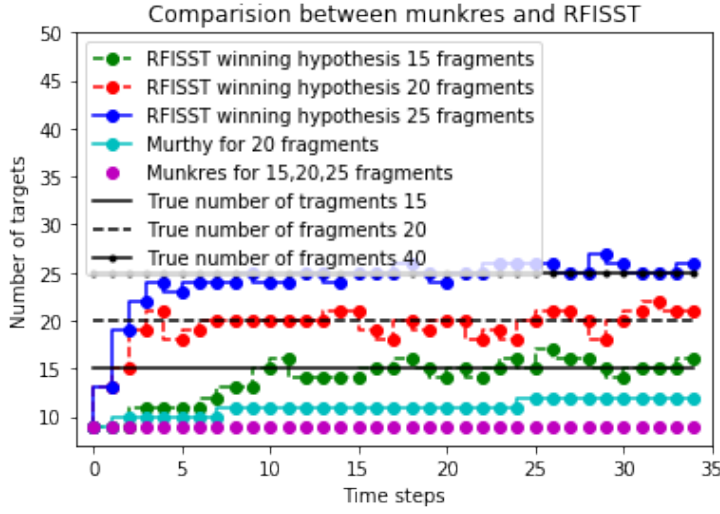


Fig. 6: Varying number of true targets to be tracked

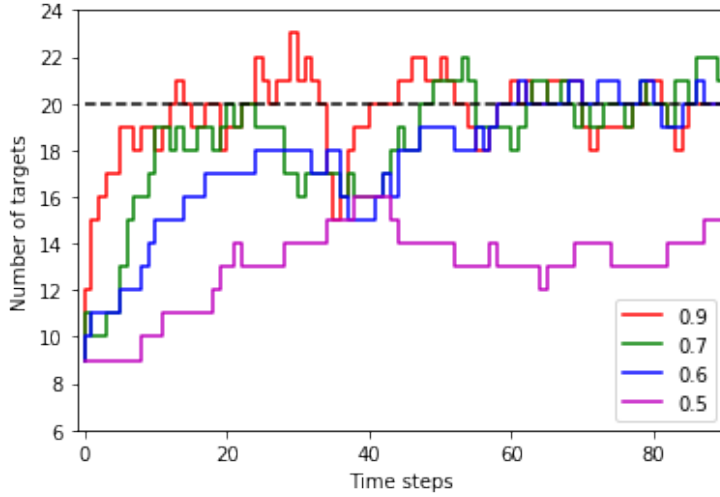


Fig. 7: Varying new birth arrival rate

A very important observation can be made from fig. 6 about Munkres assignments. It can be seen that the best hypothesis chosen by Munkres algorithm is never able to initialize new targets. This is because of the fact that we mis-model the truth using Poisson process with λ_B and λ_C . We set $\lambda_B < \lambda_C$ forcing the Munkres algorithm to always assign the observation of a new, so far unknown, fragment to clutter instead of initializing it as a new target. The Munkres algorithm is not robust to our inability to model the stochastic process underlying the arrival rates of new targets and clutter while R-FISST based filter is robust

as shown by its ability to initialize new targets. Another important observation to be made from the same figure is that R-FISST (maintaining 10 hypotheses at each time step) is much quicker than Murthy's m-best ($m=10$) in converging to the right number of targets, in fact, as can be seen, MHT with Murty is able to converge to only about 30% (3 out of 10) of the unknown fragments in the allotted time.

Finally looking at Fig. 4 which has likelihood on its y-axis and time on its x-axis. We see that on multiple instances, less likely hypothesis at a particular time step spawn the top hypothesis at the next time step. This gives further credence to the argument that we need to maintain multiple hypotheses for scenarios where new targets come into view.

5 Applications to Collisional Cascading Scenario

In this section, two SSA applications with increasing complexity are presented to demonstrate the scalability of SMC technique. The first application considers a fragmentation within a high density region of objects. The second application is a simulation of Collisional Cascading (a.k.a the Kessler Syndrome). These scenarios cause very large numbers of possible data association hypotheses. In the smallest case considered, the minimum number of association hypotheses is on the order of $11!$. Even this number of possible hypotheses is very difficult to generate exhaustively and is intractable to implement in reasonable time on standard computational resources. The goal of this section is to show that the R-FISST technique is scalable to such problems.

5.1 Fragmentation With Background Objects

In this section, a simulation is performed to begin to scale the R-FISST technique to highly ambiguous scenarios. The simulation parameters are as follows:

- Twenty-one initial near-GEO RSO.
- RSO are purposely closely spaced to cause initial ambiguity. To generate the closely spaced objects valid orbits were sampled from an initially GEO orbit. Since this is done at random this could result in an initial assignment problem with $21!$ possible associations even before fragmentation.
- General Mission Analysis Tool (GMAT) is used to propagate the orbits using the RungeKutta89 integrator with accuracy $10E12$ and a high-fidelity force model which included JGM-2 Earth potential with twenty-first degree and order. Sun, Jupiter, and lunar Point-mass gravity affect, solar radiation pressure, and atmospheric drag were also included.
- During tracking, a single fragmentation event occurs resulting in 11 fragments and was simulated using NASA's standard breakup model [13] with RSO initial characteristic length of 10 meters. Only fragments with characteristic length greater than 0.7 meters are considered.
- Object states included object's position and velocity in J2000 ECI.
- A single sensor, located near Maui, HI, is used and assumed to have a twelve degree FOV centered at the mean position of the fragmentation event.
- Fragment pdfs are initialized assuming a full state measurement.

- Observations are received in 30 second intervals containing ECI J2000 right ascension, declination, and range information. With assumed zero-mean Gaussian noise with standard deviation of 2 arc seconds in right ascension and declination and 2 km in range.
- R-FISST is initialized with a single hypothesis with Gaussian pdfs on all twenty-one objects. Each spatial distribution initial mean equal to the RSO true initial position with 10 km in position standard deviation and 0.1 km/s standard deviation in velocity.
- the simulation is stopped after one hour's worth of measurements are received.
- Assumed model parameters $\lambda_B = 1$, $p_s = 0.9$, $\lambda_C = 1$, and $p_d = 0.85$.
- No fragmentation model was assumed in the motion model.

In this example, the fragmentation occurs at time step 525. It is important to note that, for this simulation, the expected number of ambiguous measurement returns after fragmentation is typically 11, taking into account possible missed detections, false alarms and other ambiguous objects in the field of view. To add perspective, an 11 object by 11 observation data association scenario causes a minimum of 11! possible data association hypotheses, not including missed detection, false alarm, birth, and death associations. Keep in mind that the fragments tend to pass near or through the blast point every orbital period. For this reason, the data association problem does not ever completely disambiguate. Figure 8 shows the estimated number of objects versus time using a condensed box and whiskers plot at every time step. The true number of objects before the fragmentation is twenty-one and after the fragmentation is thirty-one. Even with the large inherent data association problem the R-FISST technique provides an accurate representation of the number of objects present. Most importantly it does this within an hour of event. This is important for providing timely tracking data that can prevent collisional cascading. Some other tracking techniques must wait hours for data to disambiguate before they can be applied [14].

Figure 9 shows the average total entropy throughout time. The differential entropy of a multivariate Gaussian with covariance, Σ_i , is expressed,

$$E_i = \frac{1}{2} \ln(|(2\pi e)^k \Sigma_i|) \quad (27)$$

where the bars represent the matrix determinant and k is the dimension of the multi-variate Gaussian. The average total entropy is the sum over all objects, i . After breakup the entropy of the system is high due to the uncertainty in the initial birth pdf. After the first update the entropy reduces and remains low. The entropy however has a slight upward trend. This is due to the fact that as the time from breakup increases some objects escape from the field of view. The corresponding pdfs of the objects no longer being detected slowly increase in uncertainty due to process noise.

One of the Monte Carlo simulations was used to show how estimated tracks develop over time, figures 10 and 13 show a subset of the estimated tracks plotted against the corresponding true tracks at 15 and 60 minutes respectively. At each segment, the currently observed objects are shown using red diamonds. It can be seen in 10 that some of the true objects are yet to be estimated. As time passes, more and more of the true objects are estimated. It can be seen in figure 13 that objects that receive frequent measurements lie directly over their corresponding

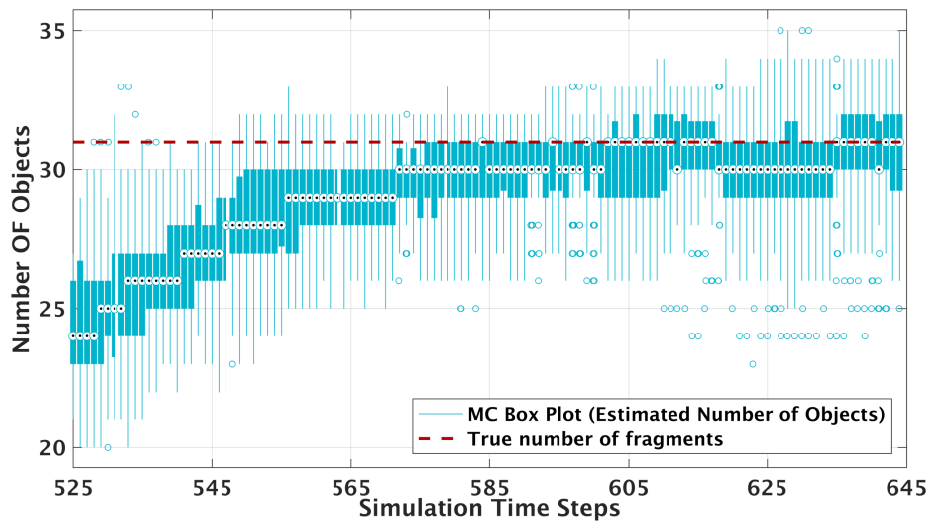


Fig. 8: Estimated number of objects throughout the simulation using a condensed box plot at each time step for the 55 Monte Carlo runs

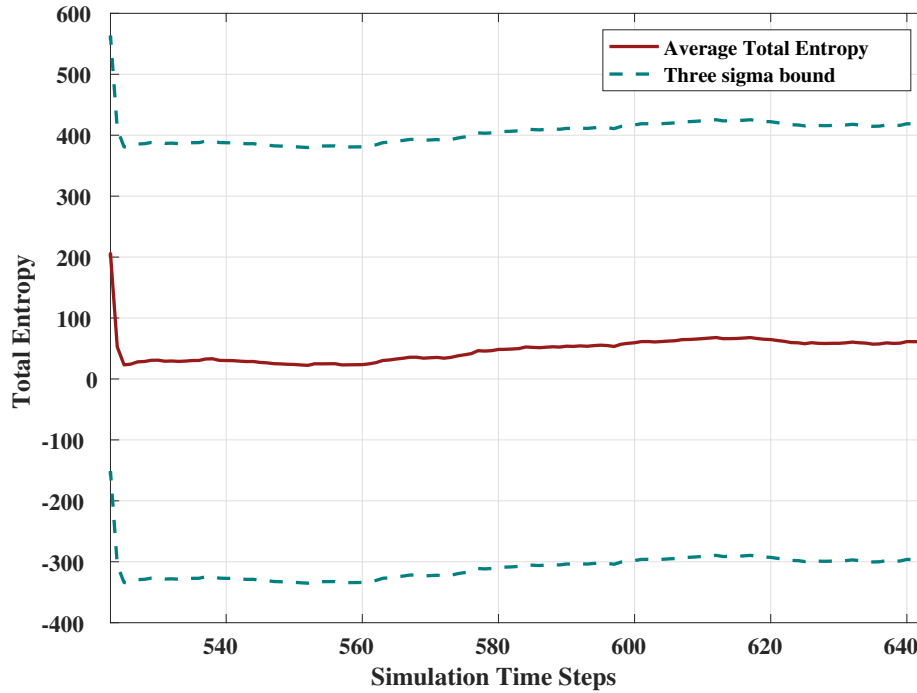


Fig. 9: The total entropy averaged over 55 Monte Carlo runs.

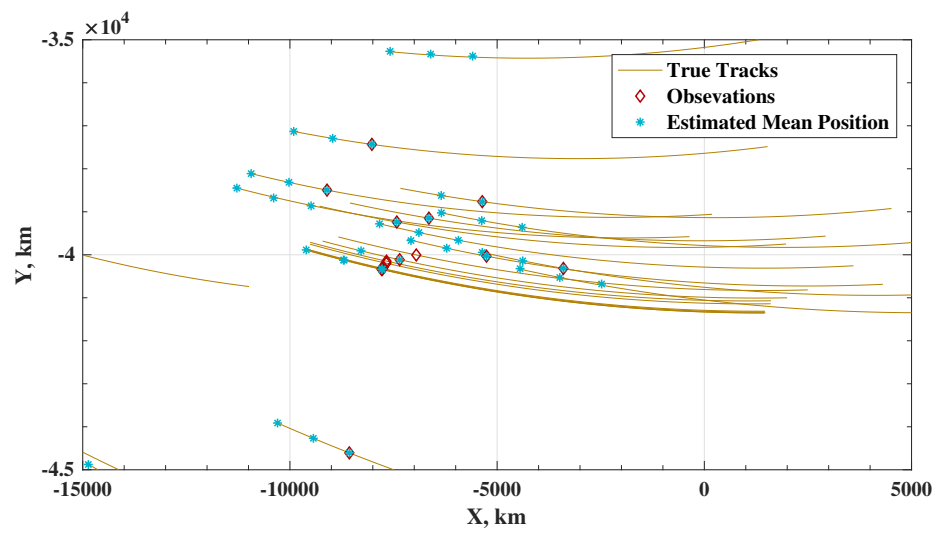


Fig. 10: Estimated x,y positions 15 minutes after breakup represented as blue stars. Estimated states are overlaid on top of true tracks

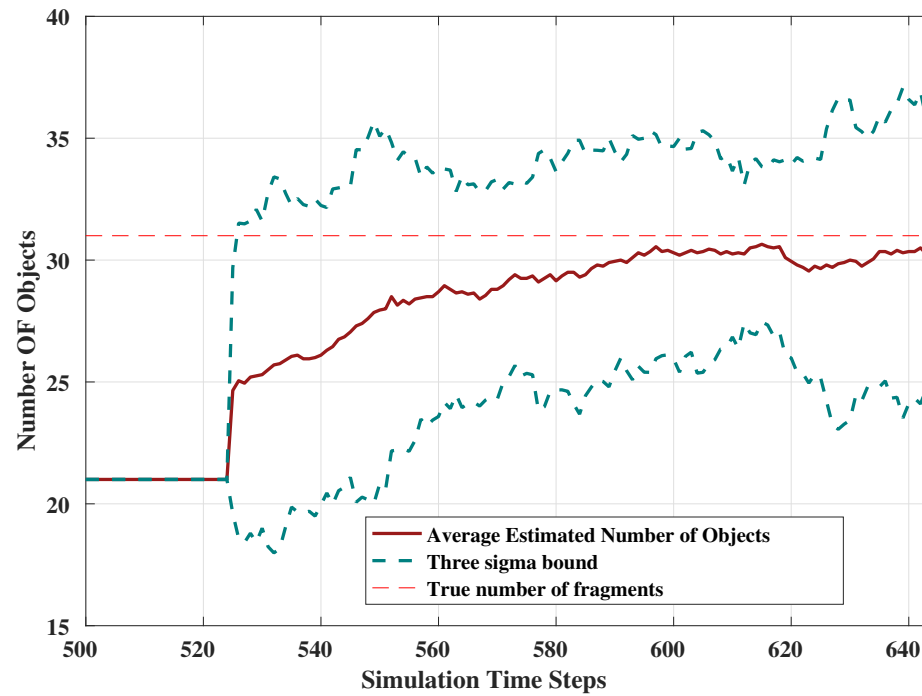


Fig. 11: The estimated number of objects throughout time averaged over 55 Monte Carlo runs

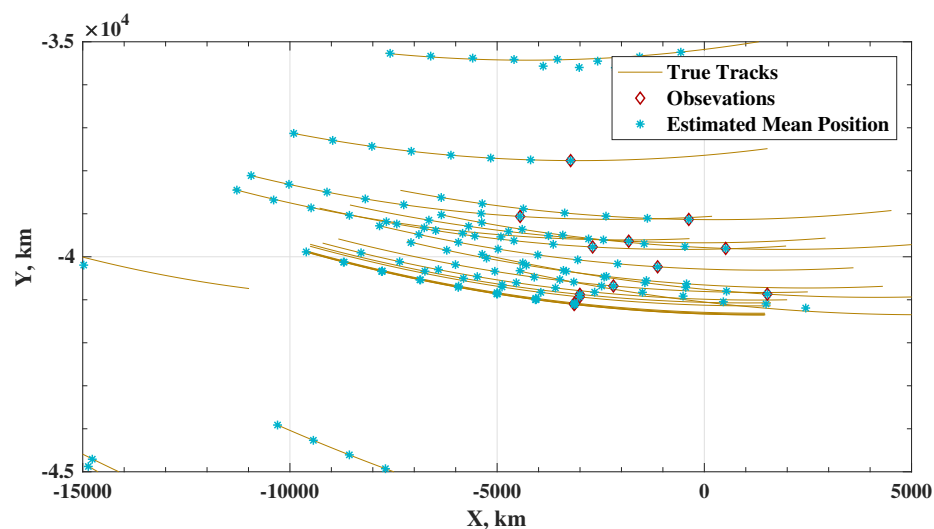


Fig. 12: Estimated x,y positions 35 minutes after breakup represented as blue stars. Estimated states are overlaid on top of true tracks

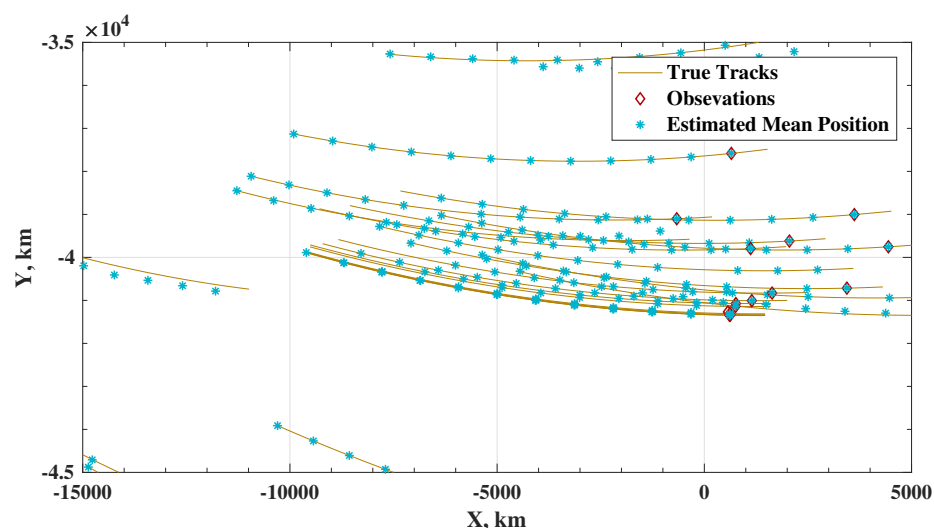


Fig. 13: Estimated x,y positions 1 hour after breakup represented as blue stars. Estimated states are overlaid on top of true tracks

true track. However, those objects that are not updated frequently may drift away from the truth until update. These deviations can cause the localization errors seen in 16, explained later. Figure 14 shows the full set of estimated tracks and observations over the entire simulation time. In this figure true tracks are also shown, but they are completely covered by estimated tracks.

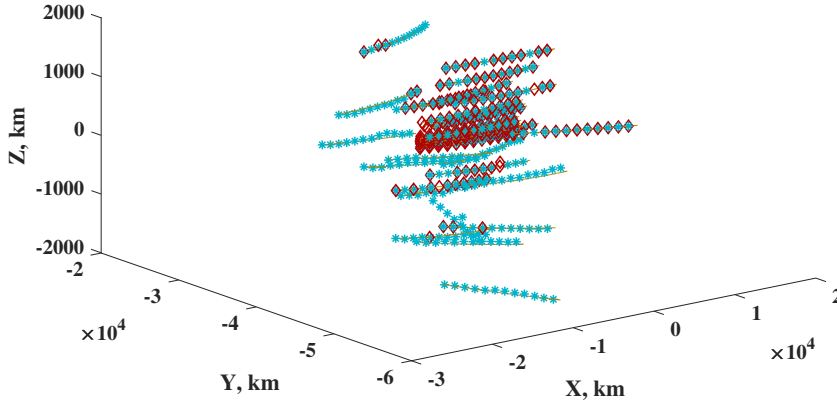


Fig. 14: Estimated 3-space tracks (blue) and observations (red diamond)

Figures 15-16 are the cardinality focused and localization focused Optimal Sub Pattern Assignment metric (OSPA)[20]. The OSPA metric is a convenient unified metric to judge the performance of multi-object tracking algorithms since it provides metrics on how well algorithms are tracking the object states as well as the number of objects, since both aspects are important in these problems. Very briefly, $OSPA = (d_{loc}^p + d_{card}^p)^{\frac{1}{p}}$, assuming $m \leq n$, $d_{loc} = \{\frac{1}{n} \sum_{i=0}^m d_c^p(x_i, y_{\pi(i)})\}^{\frac{1}{p}}$, where p is the order of the OSPA metric, d_c is the cutoff-based distance, $y_{\pi(i)}$ is the track assigned to truth x_i , $d_c(x, y) = \min\{d_b(x, y), c\}$, and $d_{card} = \{\frac{n-m}{n} c^p\}^{\frac{1}{p}}$. Larger values of cut off parameter c tend to emphasize cardinality errors and vice versa. The order parameter, p , controls the penalty assigned to target estimates that are not close to any of the ground truth tracks. These figures are included to show the performance of cardinality and state estimation in a highly ambiguous scenario. Cardinality errors decrease over time as more and more of the fragments are captured. Localization errors remain low due to the assumption of a full state measurement for fragment pdf initialization. It is important to note that this would not be the case if the fragment pdfs were initialized using techniques that can initialize using partial state measurements. In those cases, the initial localization error would be high and tend to be refined as more information is collected.

5.2 Collisional Cascading

To show full scalability of the R-FISST technique an SSA example scenario was simulated of the feared Collisional Cascading event, otherwise known as the Kessler Syndrome. In this simulation, all of the current unclassified GEO objects were taken from the GEO catalog [1] and propagated to a time in the near future. Then, over a period of ten hours, ten adjacent RSO undergo simulated cascading fragmentation events. The simulation parameters are listed as follows.

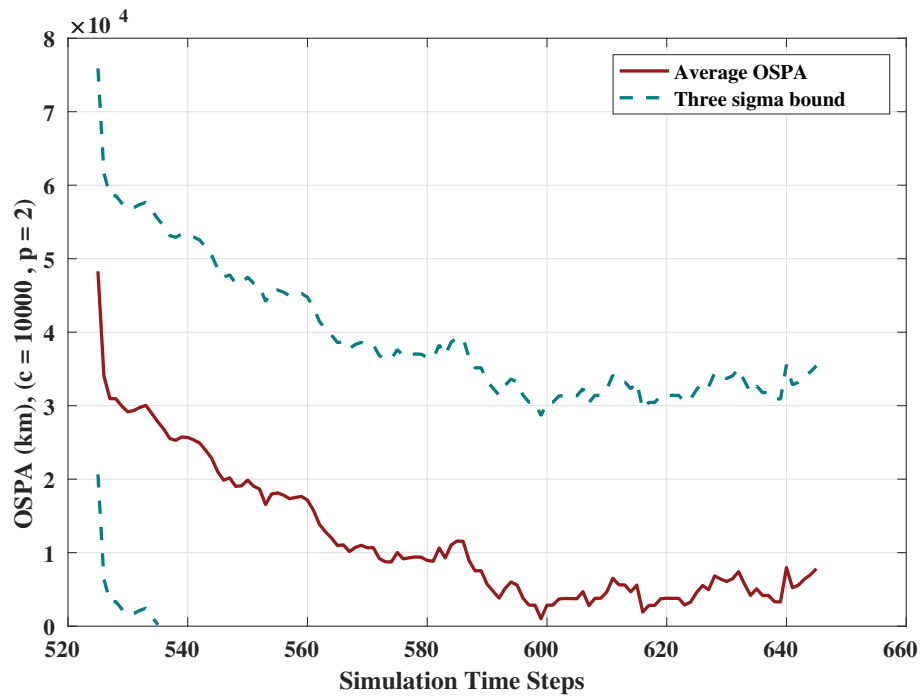


Fig. 15: The "per-object" average OSPA distance versus time. Cut-off value, $c = 10000$ km to emphasize cardinality errors. (averaged over 55 Monte Carlo simulations)

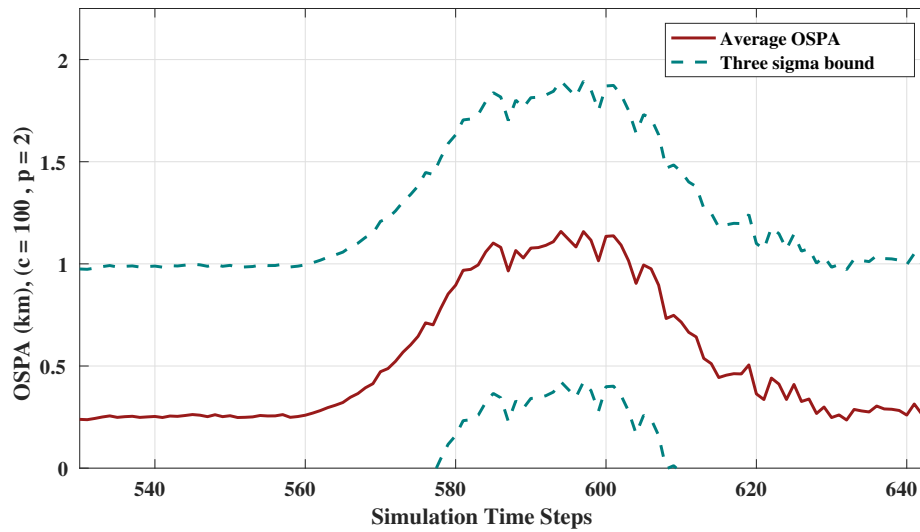


Fig. 16: "per-object" average OSPA distance versus time. Cut-off value, $c = 100$ km to emphasize localization errors. (averaged over 55 Monte Carlo simulations)

- Initially the entire current GEO population was taken from the GEO public catalog. Ten fragmentations occur resulting in a new population of 952 objects. Signifying over 100 fragments were created from the collisional cascading.
- Each fragmentation event was simulated using NASA’s standard breakup model [13] with RSO initial characteristic length of 10 meters. Only fragments with characteristic length greater than 0.7 meters are considered.
- Objects were tracked for up to 37.33 hours using the R-FISST technique.
- Object states included object’s position and velocity in J2000 ECI.
- It is assumed that there is full coverage of the GEO population. Position measurements are provided with observation noise considered to be Normally distributed with standard deviation $\sigma_p = 100\text{m}$.
- Fragment pdfs are initialized assuming a full state measurement.
- Observations are received in 10 minute intervals.
- R-FISST is initialized with a single hypothesis with Gaussian pdfs on all of the initial GEO objects. Each with spatial distribution initial mean equal to the RSO true initial position with 10 km in position standard deviation and 0.1 km/s standard deviation in velocity.
- Assumed model parameters $\lambda_B = 1$, $p_s = 0.9$, $\lambda_C = 1$, and $p_d = 0.9$.
- No fragmentation model was assumed in the motion model.

The goal of this simulation is to show that the R-FISST technique can maintain accurate tracking performance even in worst case scenarios like Collisional Cascading. Two cases are compared. Both cases are under the same assumptions. However, in case one, the simulated observations did not contain any missed detections or false alarms. In this case, as soon as measurements pertaining to the fragments are received, the estimated number of objects converges to the true number of objects. In the second case, missed detections are allowed and false alarms are received. In this case it takes much longer to converge to the true number of objects. This is illustrated in figure 17. Even though there is a delay all fragments are captured within 38 hours. R-FISST is capable of capturing these objects much faster than today’s techniques because of it’s ability to keep the problem computationally tractable[8]. As expected, performance will be adversely affected under real-world operational constraints. Total entropy over time is shown in figure 18.

6 Conclusion

This manuscript has discussed the R-FISST technique for multi-object tracking. R-FISST allows one to keep the multi-object tracking problem computationally tractable through the use of the MCMC method. The manuscript illustrates the scalability of R-FISST to space object fragmentation events. It also touches upon the tuning parameters available and details the advantages inherent in the R-FISST approach when compared to an MHT implementation with GNN heuristic. Simulations detailing the robustness of the R-FISST filter were shown using localization and cardinality errors. The results show that the R-FISST technique is able to keep the problem computationally tractable and provide timely and accurate tracking. Future work includes adapting the development to be used on high performance computers and include maneuver models.

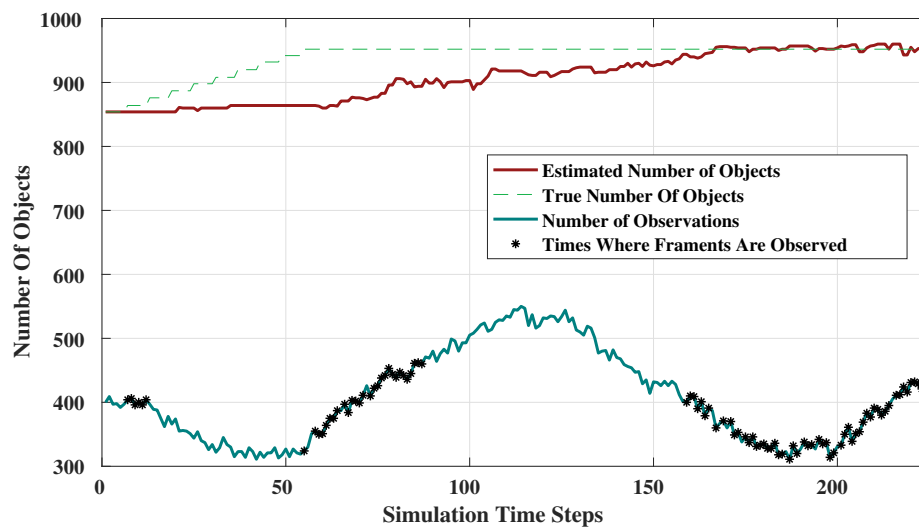


Fig. 17: The number of estimated objects for the Collisional Cascading simulation with missed detections and false alarms

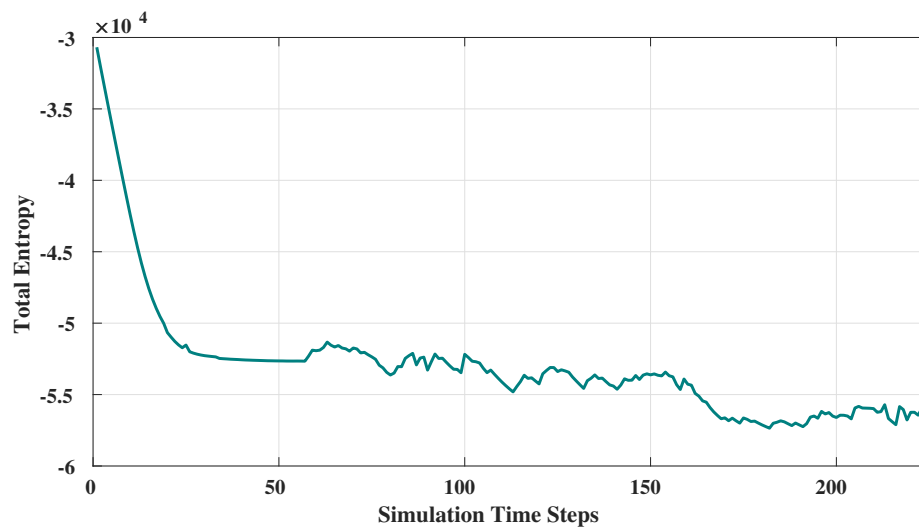


Fig. 18: The total entropy throughout time for the Collisional Cascading simulation with missed detections and false alarms.

Conflict Of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

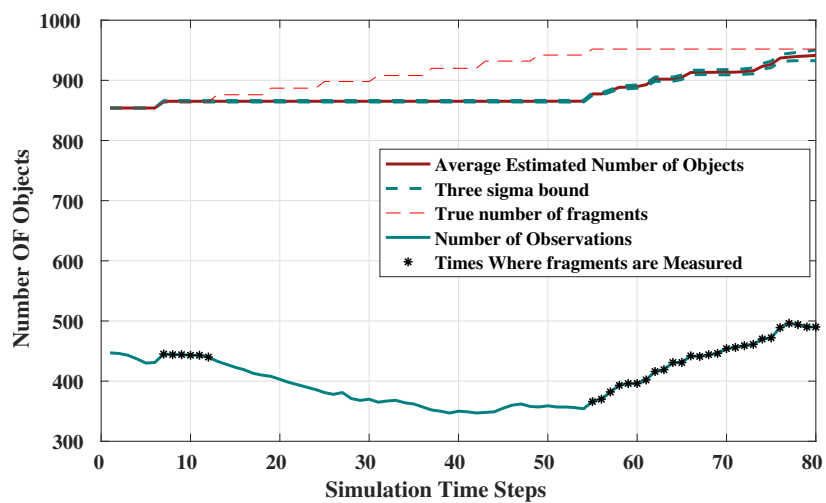


Fig. 19: The number of estimated objects for the Collisional Cascading simulation with no missed detections or false alarms.

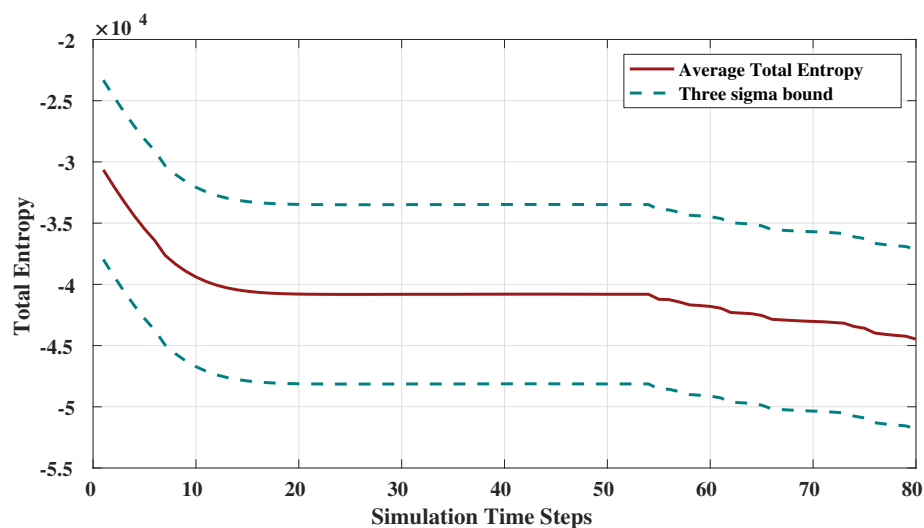


Fig. 20: The total entropy throughout time for the Collisional Cascading simulation with no missed detections or false alarms.

Acknowledgment

This work was funded by AFOSR grant number: FA9550-13-1-0074 under the Dynamic Data Driven Application Systems (DDDAS) program.

References

1. Resident space object catalog <https://www.space-track.org/>. Web (2017)
2. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, San Diego, CA (1988)
3. Bergman, N., Doucet, A.: Markov chain monte carlo data association for target tracking. In: Proc. ICASSP (2000)
4. Blackman, S.: Multiple hypothesis tracking for multiple target tracking. IEEE Tr. Aerospace and Electronic Systems **19**, 5–19 (2004)
5. Chakravorty, S., Faber, W.R., Hussein, I.I., Mishra, U.R.: A Belief Space Perspective of RFS based Multi-Target Tracking and its Relationship to MHT. ArXiv e-prints
6. Cong, S., et al.: Markov chain monte carlo approach for association probability evaluation. Proc. IEE **151**, 185–193 (2004)
7. Faber, W., Chakravorty, S., Hussein, I.: A randomized sampling based approach to multi-object tracking. In: Proceedings of the 18th International Conference on Information Fusion, pp. 1307–1314. IEEE, Piscataway, NJ (2015)
8. Faber, W., Chakravorty, S., Hussein, I.: A randomized sampling based approach to multi-object tracking with comparison to HOMHT. Advances in the Astronautical Sciences **156** (2015). (Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Vail, CO, August 9–13 2015, Paper AAS 15-745)
9. Faber, W., Chakravorty, S., Hussein, I.: R-FISST and the data association problem with applications to space situational awareness. In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference. Long Beach, CA (2016)
10. Faber, W., Zaidi, W., Hussein, I., Roscoe, C.W.T., Wilkins, M.P., Paul W. Schumacher, J.: Application of multi-hypothesis sequential monte carlo for breakup analysis (2017). (Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, Paper AAS 17-579)
11. Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE J. Oceanic Eng. pp. 173–184 (1983)
12. Hoang, H., Vo, B.T., Vo, B.N.: A fast implementation of the generalized multi bernoulli filter with joint prediction and update. In: International Conference on Information Fusion (Fusion) (2015)
13. Johnson, N.L., Krisko, P.H., Lieu, J.C., Am-Meador, P.D.: NASA’s new breakup model of evolve 4.0. Adv. Space Res. **Vol. 28, No. 9** (2001). Pp. 1377-1384
14. Kelec, T., Shoemaker, M., Jah, M.: Application of the constrained admissible region multiple hypothesis filter to initial orbit determination of a break-up. Proceedings of the 6th European Conference on Space Debris Darmstadt, Germany (08/2013)
15. Mahler, R.P.S.: Statistical Multisource-Multitarget Information Fusion. Artec House (2007)
16. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957)
17. Oh, S., Russell, S., Sastry, S.: Markov chain monte carlo data association for mutli-target tracking. IEEE Transactions on Automatic Control **54**(3), 481–497 (2009)
18. Pasula, H., et al.: Tracking many objects with many sensors. In: Proc. IJCAI (1999)
19. Reid, D.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control **24**(6), 843–854 (1979)
20. Schuhmacher, D., Vo, B.T., Vo, B.N.: A consistent metric for performance evaluation of multi-object filters. IEEE Transactions On Signal Processing **56** (8), 3447–3457 (2008)
21. Stauch, J., Bessell, T., Rutten, M., Baldwin, J., Jah, M., Hill, K.: Joint probabilistic data association and smoothing applied to multiple space object tracking. Journal of Guidance, Control, and Dynamics **41**(1), 19–33 (2018). DOI 10.2514/1.g002230
22. Vo, B.N., et al.: Labeled random finite sets and multi-object conjugate priors. IEEE Transactions on Signal Processing **61**, 3460–3475 (2013)
23. Vo, B.N., et al.: Labeled random finite sets and the bayes multi-target tracking filter. IEEE Transactions on Signal Processing **62**, 6554–6567 (2014)
24. Vo, B.T., et al.: An analytic implementation of the cardinalized probability hypothesis density filter. IEEE Tr. Signal Processing **55**, 3553–3567 (2007)

Particle Gaussian Mixture Filters-I[★]

Dilshad Raihan, Suman Chakravorty

Department of Aerospace Engineering, Texas A & M University, College Station, TX, 77840

Abstract

Recursive estimation of nonlinear dynamical systems is an important problem that arises in several engineering applications. It is well known that the posterior state estimates in nonlinear problems may assume non-Gaussian multimodal densities. In the past, Gaussian mixture filters and particle filters were introduced to handle non-Gaussianity and nonlinearity. However, these methods have found only limited success as most Gaussian mixture filters attempt to fix the number of mixture modes during the estimation process, and the particle filters suffer from the curse of dimensionality. In this paper, we propose a particle based Gaussian mixture filtering approach for nonlinear estimation that is free of the particle depletion problem inherent to most particle filters. We employ an ensemble of possible state realizations for the propagation of state probability density. A Gaussian mixture model (GMM) of the propagated uncertainty is then recovered by clustering the ensemble. The posterior density is obtained subsequently through a Kalman measurement update of the mixture modes. We prove the convergence in probability of the resultant density to the true filter density assuming exponential forgetting of initial conditions by the true filter. The performance of the proposed filtering approach is demonstrated through several test cases and is extensively compared to other nonlinear filters. The particle Gaussian mixture filter is capable of handling the non-Gaussianity of the state pdf arising from dynamics, initial conditions or process noise. A more general approach that can also handle non-Gaussianity related to measurement update is presented in a companion paper.

Key words: Estimation Algorithms; Nonlinear Filters; Gaussian mixture models; Multimodality; Particle Filtering; State Estimation; Machine Learning; Curse of Dimensionality; Kalman Filters.

1 Introduction

Rapid advances in the fields of control and automation has inspired a significant interest in recursive and computationally efficient algorithms for estimating the state and associated uncertainty in higher dimensional nonlinear systems. The Kalman filter provides the unbiased minimum variance estimator for linear dynamical systems perturbed by additive Gaussian noise [19, 20]. The extended Kalman filter (EKF) was introduced to incorporate nonlinear systems into the Kalman filtering framework [29]. However, the limitations of the Jacobian linearization assumptions and the accumulation of linearization errors can result in the divergence of EKF estimates. The Unscented Kalman Filter (UKF) and the broader class of sigma point Kalman filters provided a

derivative free alternative to the EKF [17, 34]. Both EKF and UKF approximate the posterior pdf with a single Gaussian pdf. However, the state pdf in a general nonlinear filtering problem can be non-Gaussian and multimodal. Cubature Kalman filters (CKF) that rely on a spherical-radial cubature rule to evaluate the integrals involved in the estimation have been proposed [4]. A variant of cubature filters, that perform random scaling and rotation of cubature points and axes, known as stochastic integration filter (SIF) has also been proposed recently [11]. A Gaussian mixture approximation of the state pdf was proposed to incorporate the multimodality of the problem in nonlinear settings [2, 31]. These approaches however had a major shortcoming as the number of Gaussian components were fixed initially and kept constant through out the estimation process. Also the component weights were updated only during the measurement update. Approaches to adapting the weights of individual Gaussian modes by minimizing the propagation error committed in the Gaussian mixture model (GMM) approximation have been proposed recently [32]. A different approach to improving the accuracy of GMM filters is by splitting the Gaussian components during the propagation based on non-linearity induced distortion [9]. Both of these approaches

[★] This paper was not presented at any IFAC meeting. This work was supported by the AFOSR grants FA9550-13-1-0074, FA9550-17-1-0068 under the Dynamic Data Driven Application Systems (DDDAS) program and the NSF award ECCS1637889. Corresponding author Suman Chakravorty. Tel. 979-458-0064

Email addresses: dilshadraihan@tamu.edu (Dilshad Raihan), schakrav@tamu.edu (Suman Chakravorty).

require frequent optimizations, or entropy calculations, to be performed during the propagation, which significantly add to the overall computational requirement. A Gaussian mixture ‘blob’ filter that relies on EKF for propagation and update has been proposed recently [26]. It performs a resampling step between the propagation and update stages. The resampling step ensures that the component covariances of the propagated pdf all obey a linear matrix inequality (LMI) based upper bound.

The particle filters (PF) are a class of sequential Monte Carlo methods that employ an ensemble of possible state realizations known as particles to represent the state pdf [5, 15]. These states are sampled from the initial pdf and propagated forward in time based on the nonlinear system model. The measurement updates are performed by assigning weights to individual particles which may then be resampled. The PF does not enforce restrictive assumptions on the nature of dynamics or pdf. However Particle filters are subject to the curse of dimensionality due to the particle depletion problem wherein a significant fraction of particles lose their importance weights during the measurement update. Preventing depletion requires the number of particles to be increased exponentially with the dimension of state space [6]. Particle based approaches such as the Ensemble Kalman filter (EnKF) and the Feedback particle filter (FPF) that forego the resampling based measurement update have been demonstrated to be effective in higher dimensional filtering problems involving unimodal pdfs [12, 35]. The Gaussian sum particle filter (GSPF) is a nonlinear filter that uses a GMM representation of the state pdf [21]. It obtains an ensemble of particles from each GMM component. The ensembles are then propagated forward separately like a parallel bank of filters. The GSPF relies on an importance sampling based approach to perform the measurement update.

In this paper, we propose a particle Gaussian mixture (PGM) filter for nonlinear estimation. The PGM filter design is inspired by a previous work on a UKF-PF hybrid filter that was proposed for space object tracking [27]. The PGM filter employs an ensemble of possible state realizations for performing the uncertainty propagation. A functional form of the propagated pdf is recovered as a GMM by clustering the particles. The posterior pdf is obtained by performing a Kalman measurement update on the GMM. The PGM filter is conceived to keep track of the nonlinear uncertainty propagation, without performing any additional optimization or splitting operations during the propagation step. As the posterior pdf is obtained without employing the particle measurement update, the PGM filter is not prone to the particle depletion problem and the associated curse of dimensionality. Since the additional clustering step is performed only during the measurement update step, the PGM filter is especially suitable for filtering in the sparse measurement scenario.

The remainder of this article is organized as follows. An introductory discussion on mixture model filtering is given in section 2. The PGM filter algorithm, and an associated convergence result, are presented in section 3. Details pertaining to the actual implementation of the proposed filter are given in section 4. The PGM filter is applied to three test cases and compared extensively with other nonlinear filters in section 5.

2 Preliminaries: Mixture Model Filtering

Let the state of the dynamical system of interest be denoted by $x \in \mathbb{R}^d$. We assume that the state of the system evolves according to a Markov chain whose transition density is known and is specified by $p_n(x'|x)$. We also obtain measurements of the state at discrete times n and the observation model is specified by the following:

$$z_n = h(x_n) + v_n, \quad (1)$$

where $h(x)$ is a nonlinear measurement function and $\{v_n\}$ is a discrete time Gaussian white noise process with zero mean and covariance R_n . Let $\pi_{n-1}(x)$ represent the conditional state pdf $p_{n-1}(x|Z_{n-1})$ where Z_{n-1} is the sequence of all measurements recorded until time $n-1$. Additionally, let the prediction of the pdf before the measurement z_n at time n (the predicted prior pdf) $p_n(x|Z_{n-1})$ be represented by $\pi_n^-(x)$. Then :

$$\pi_n^-(x) = \int p_n(x|x')\pi_{n-1}(x')dx'. \quad (2)$$

Further, after measurement z_n is received, the posterior pdf of the state is obtained according to Bayes rule as:

$$\pi_n(x) = \frac{p_n(z_n|x)\pi_n^-(x)}{\int p_n(z_n|x')\pi_n^-(x')dx'}, \quad (3)$$

$p_n(z_n|x)$ is the measurement likelihood function and can be inferred from the measurement model above. The prediction and the update steps above are the key steps to any recursive filtering algorithm. Let us assume that a mixture representation has been chosen for the predicted and posterior pdfs. In particular, let:

$$\begin{aligned} \pi_n^-(x) &= \sum_{i=1}^{M^-(n)} \omega_i^-(n) p_{i,n}^-(x), \\ \pi_n(x) &= \sum_{i=1}^{M(n)} \omega_i(n) p_{i,n}(x), \end{aligned} \quad (4)$$

where $p_i^-(\cdot), p_i(\cdot)$ are pdfs, and $\{\omega_i^-(n)\}, \{\omega_i(n)\}$ are positive sets of weights that both add up to unity. The terms $M^-(n)$ and $M(n)$ represent the number of components used in the mixture representation. The prediction equation for the mixture model then boils down to

the following:

$$\pi_n^-(x) = \sum_{i=1}^{M(n-1)} \underbrace{\omega_i^-(n)}_{\omega_i^-(n)} \underbrace{\int p_n(x|x') p_{i,n-1}(x') dx'}_{p_{i,n}^-(x)}. \quad (5)$$

Explicitly, the mixture prediction step can be split into the following two steps:

$$\omega_i^-(n) = \omega_i^-(n-1), \quad (6)$$

$$p_{i,n}^-(x) = \int p_n(x|x') p_{i,n-1}(x') dx'. \quad (7)$$

Given an observation z_n , the prior mixture $\pi_n^-(x)$ is transformed into the posterior mixture $\pi_n(x)$ as follows:

$$\pi_n(x) = \frac{\sum_{i=1}^{M^-(n)} \omega_i^-(n) p_n(z_n|x) p_{i,n}^-(x)}{\sum_{i=1}^{M^-(n)} \omega_i^-(n) \int p_n(z_n|x') p_{i,n}^-(x') dx'}. \quad (8)$$

Define the likelihood that z_n comes from the i^{th} mixture component as:

$$l_i(n) \equiv \int p_n(z_n|x') p_{i,n}^-(x') dx'. \quad (9)$$

Rearranging the above mixture expression using the definition of the component/ mode likelihood gives us:

$$\pi_n(x) = \sum_{i=1}^{M^-(n)} \underbrace{\frac{\omega_i^-(n) l_i(n)}{\sum_j \omega_j^-(n) l_j(n)}}_{w_i(n)} \underbrace{\frac{p_n(z_n|x) p_{i,n}^-(x)}{l_i(n)}}_{p_{i,n}(x)}. \quad (10)$$

The above expression shows that the measurement update has a hybrid nature, a standard update of the individual modes of the mixture with the measurement z_n , and a discrete Bayesian update of the mode weights using the mode likelihoods $l_i(n)$. Note that the mode likelihoods are the Bayes' normalization factors for the individual modes. Explicitly, we delineate the discrete and continuous updates of the mixture model below:

$$\omega_i(n) = \frac{\omega_i^-(n) l_i(n)}{\sum_j \omega_j^-(n) l_j(n)}, \quad (11)$$

$$p_{i,n}(x) = \frac{p_n(z_n|x) p_{i,n}^-(x)}{l_i(n)}. \quad (12)$$

Let us now assume that we have fixed the form of the mixture model to a GMM, i.e., the posterior pdf at time

$n-1$ can be represented by the GMM:

$$p_{i,n-1}(x) = \mathcal{G}(x; \mu_i(n-1), P_i(n-1)), \quad (13)$$

where $\mathcal{G}(x; \mu, P)$ represents the Gaussian pdf with mean μ and covariance P . Consider first the prediction equations. Note that from (5), the number of mixture components at time $n-1$, $M(n-1)$, is the same as the number of mixture components of the prediction at time n , $M^-(n)$. However, this assumes that the prediction of the i^{th} Gaussian component $p_{i,n-1}$ of the posterior pdf at time $n-1$ remains a single Gaussian at time n , $p_{i,n}^-$. However, this is, in general, not true. The number of mixture components necessary to approximate the state pdf may vary from one time step to the other. For example, consider the nonlinear dynamical system given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{x_1}{2} \\ \sin(\frac{x_2}{2}) \end{bmatrix} + \Gamma(t), \quad (14)$$

$$\pi_0(X) = \mathcal{G}(X, \begin{bmatrix} -12 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.2 & 0 \\ 0 & 1 \end{bmatrix}), \quad (15)$$

where $\Gamma(t)$ is a white noise process. In Fig.1a, the locations of 200 particles sampled from the unimodal initial pdf $\pi_0(x_0)$ are seen to separate into two distinct modes as time progresses. Hence, in order to use mixture models for prediction, we have to find a way to deal with time varying number of GMM components.

Next, let us consider the measurement update (12). Since the prior component is Gaussian, and if the update (12) is approximated using the Kalman/ linear minimum mean squared error (LMMSE) update [22,27], we have :

$$\mu_i(n) = \mu_i^-(n) + P_{i,zx}^T(n) P_{i,zz}^{-1}(n) (z_n - E_i[h(X)]), \quad (16)$$

$$P_i(n) = P_i^-(n) - P_{i,zx}^T(n) P_{i,zz}^{-1}(n) P_{i,zx}(n), \quad (17)$$

where

$$P_{i,zx}(n) = E_i \left[\left(h(X) - E_i(h(X)) \right) \left(X - E_i(X) \right)^T \right], \quad (18)$$

$$P_{i,zz}(n) = E_i \left[\left(h(X) - E_i(h(X)) \right) \left(h(X) - E_i(h(X)) \right)^T \right] \quad (19)$$

and $E_i[f(X)]$ represents an expectation of the function $f(X)$ with respect to the random variable X where $X \sim \mathcal{G}(x; \mu_i^-(n), P_i^-(n))$. However, similar to the prediction case, in general, a single predicted Gaussian component can split into multiple modes after the update 12. An illustration of this is given in Fig. 1b. In this case we have a prior ensemble generated from $\pi(x) =$

$\mathcal{G}(X, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix})$. Then, a noisy measurement $z = 2$ is recorded where

$$z = x_1^2 + \tau, \quad \tau \sim \mathcal{G}(x, 0, 2).$$

An ensemble for the posterior pdf $\pi(x|z)$ is obtained through resampling and is seen to split into two separate modes. Hence, just as in the prediction step, there is a need to deal with the time varying number of GMM components after an update.

For ease of treatment and clarity of exposition, we shall not consider the measurement update aspect of the GMM filtering problem in this paper, which will be treated in a companion paper. Hence, we make the following assumption for the remainder of the paper.

Assumption 1 *We shall assume a Gaussian mixture representation for the predicted and posterior filtered densities. Further, we assume that given a predicted mixture component at time n , $\mathcal{G}(x; \mu_i^-(n), P_i^-(n))$, the update (12) after an observation z_n is approximated arbitrarily well by the Least Squares/ Kalman update (16-17).*

3 The Particle Gaussian Mixture (PGM) Filter

In this section, we first present the PGM filter. The basic assumption underlying the PGM algorithm is that

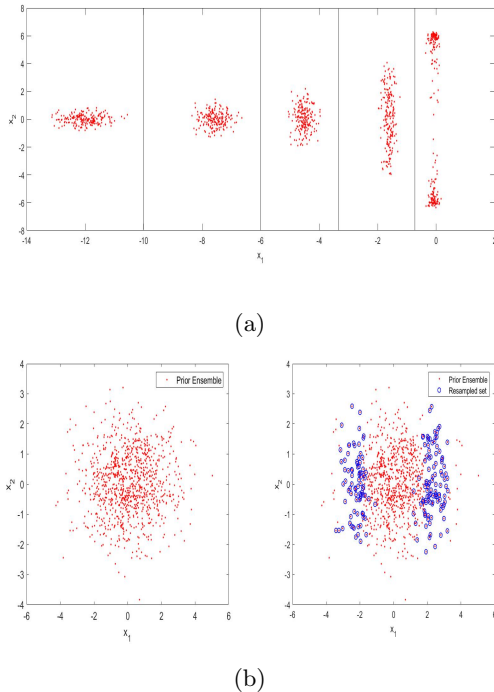


Fig. 1. Formation of multimodality a) through dynamics b) through measurement update

the predicted prior and posterior filter densities can be represented using a GMM. In particular, let:

$$\pi_n^-(x) = \sum_{i=1}^{M^-(n)} \omega_i^-(n) \mathcal{G}_i^-(x; \mu_i^-(n), P_i^-(n)), \quad (20)$$

$$\pi_n(x) = \sum_{i=1}^{M(n)} \omega_i(n) \mathcal{G}_i(x; \mu_i(n), P_i(n)). \quad (21)$$

In general, $M^-(n)$ and $M(n)$ need not be the same, however, owing to Assumption 1, they are assumed to be equal for the purposes of this paper. For instance, given a linear measurement function, this is true. The PGM filtering algorithm is composed of three basic steps that are described below.

1) Sampling: The PGM filter assumes the availability of the Markov transition kernel $p_n(x|x')$ and an efficient means of sampling from it so that one can easily draw samples of the next state x given that the current state is x' .

The first step in the PGM algorithm is the use of the transition kernel to generate a set of samples at the next time step (which is the same as in a Particle filter). In practice, we first draw an ensemble S_n of N_p states $\{x_n^1, \dots, x_n^i, \dots, x_n^{N_p}\}$ from the GMM $\pi_n(x)$ and N_p independent samples of the process noise term $w(n)$ from its density $P_W(w)$ to get $Sw_n = \{w_n^1, \dots, w_n^i, \dots, w_n^{N_p}\}$. Let

$$x_{n+1}^i = f(x_n^i) + w_n^i. \quad (22)$$

2) Clustering: Then, we use a clustering algorithm $\mathcal{C}$ to partition the set of points into $M^-(n)$ different clusters whose means and covariances can be evaluated using sample averaging. Clustering is a field of Machine learning termed as Unsupervised Learning [1, 28].

Algorithm 1 PGM Algorithm

Given $\pi_0(x_0) = \sum_{i=1}^{M(0)} \omega_i(0) \mathcal{G}_i(x_0; \mu_i(0), P_i(0))$, transition density kernel $p_n(x|x')$, $n = 1$.

- (1) Sample N_p particles $X^{(i)}$ from from π_{n-1} and the transition kernel $p_n(x|x')$ as follows:
 - (a) Sample $X^{(i)'} from $\pi_{n-1}(\cdot)$.$
 - (b) Sample $X^{(i)}$ from $p(\cdot|X^{(i)'})$.
 - (2) Use a clustering algorithm $\mathcal{C}$ to cluster the set of particles $\{X^{(i)}\}$ into $M^-(n)$ Gaussian clusters with weights, mean and covariance given by $\{\omega_i^-(n), \mu_i^-(n), P_i^-(n)\}$.
 - (3) Update the mixture weights and the mixture means and covariances to $\{\omega_i(n), \mu_i(n), P_i(n)\}$, given the observation z_n , utilizing the Kalman update (16), (17).
 - (4) $n = n+1$, go to Step 1.
-

A brief description of clustering is provided in Appendix A. In the experimental results presented in this paper, we use the simple k-means clustering algorithm [23], which is computationally very inexpensive while still being able to give good results for well separated clusters. The k-means clustering is a popular approach to partitioning wherein the data set is grouped into different clusters so that the sum of squares of within-group distances is minimized, i.e, the data set S is partitioned into M clusters $G_M^* = \{S_1, \dots, S_M\}$ such that

$$G_L^* = \underset{G_L}{\operatorname{argmin}} \sum_{i=1}^M \sum_{x_j \in S_i} \|x_j - \mu_i\|^2. \quad (23)$$

Here G_M denotes any partition of the set S into M clusters and μ_i represents the mean of the elements of the i^{th} cluster in that partition. Once the vectors x_i are assigned into different clusters, an M mode GMM describing the set S may be derived as follows.

$$n_i = \sum_{j=1}^N \mathbb{1}(x_j \in S_i), \quad w_i = \frac{n_i}{N}, \quad (24)$$

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in S_i} x_j, \quad C_i = \frac{\sum_{x_j \in S_i} (x_j - \mu_i)(x_j - \mu_i)^T}{n_i - 1}.$$

Here $\mathbb{1}(\cdot)$ represents the indicator function.

3) Measurement update: Incorporate the measurement information by updating the means and covariances of all M modes individually using a Kalman measurement update/ LMMSE update. Also update the mixture weights using the mode likelihoods $l_i(n)$ as in (11). In the present work we have considered two different approaches to computing the covariance terms $(P_{i,ZX}(n+1), P_{i,ZZ}(n+1))$ and the expectations $(E_i(h(X)))$ required for performing the Kalman update.

- (a) Update 1(PGM1-UT): In this approach, we compute the statistics of the posterior random variable with the unscented transform using a set of $2d+1$ sigma points that are distributed symmetrically. The covariance terms and the expectations required for computing the Kalman gain and posterior statistics are then computed as the weighted sample averages from the sigma points.
- (b) Update 2 (PGM1): In this approach, the covariances and cross covariances required for computing the gain matrix are evaluated directly from the particles. Let $S_{j,n+1}^{-1} = \{x_{j,n+1}^{1-}, \dots, x_{j,n+1}^{i-}, \dots, x_{j,n+1}^{Nj-}\}$ denote the set of particles that form the j -th cluster. Then the mean and covariance terms required for updating the cluster j are assigned the corresponding sample averages computed from

$S_{j,n+1}^{-1}$. The statistics of the measurement random variable are computed from the sample $Y_{j,n+1}^{-1} = \{h(x_{j,n+1}^{1-}), \dots, h(x_{j,n+1}^{i-}), \dots, h(x_{j,n+1}^{Nj-})\}$

Recursive implementation of the prediction, clustering and update steps as described here constitutes the PGM filter.

3.1 Analysis of the PGM Filter

In the following, we analyze the PGM filter. We show that under the assumption of a perfect clustering scheme $\mathcal{C}$, the PGM filter density converges in probability to the true filter density.

Let $F_{z_n}(\pi_{n-1}) = \pi_n$ denote the true filter density at time n given that the filter density at time $n-1$ is π_{n-1} and the observation at time n is z_n . Further, let $\hat{F}_{z_n}(\pi_{n-1})$ denote the filter density approximated by the PGM filter. We make the following exponential forgetting assumption on the true filter.

Assumption 2 We assume that there exists $C < \infty$ and $\rho < 1$ such that:

$$\|F_{z_n}(F_{z_{n-1}}(\dots(F_{z_1}(\pi_0))\dots)) - F_{z_n}(F_{z_{n-1}}(\dots(F_{z_1}(\pi'_0))\dots))\| \quad (25)$$

$$\leq C\rho^n \|\pi_0 - \pi'_0\|,$$

for any measurement sequence $\{z_1, z_2, \dots, z_n\}$, any π_0, π'_0 , and where $\|\cdot\|$ denotes the L_1 norm.

The conditions under which the exponential forgetting assumption holds is an active area of research [10]. However, the rate of forgetting is not always exponential. For example, parameter estimation problems do not satisfy the exponential forgetting assumption.

Let $\hat{F}_{z_n}(\hat{F}_{z_{n-1}}(\dots(\hat{F}_{z_0}(\pi_0))\dots))$ denote the filtered density approximated by the PGM filter given the measurement sequence $\{z_1, z_2, \dots, z_n\}$ and the initial density π_0 .

Assumption 3 Let $\operatorname{Prob}(\|\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})\| > \epsilon) < \delta$, for all n . Further, we assume that $\operatorname{Prob}(\|\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})\| > M) = 0$, for all n , for some $M < \infty$ (the error in a one step approximation of the filter density is almost surely uniformly bounded over all time).

The true posterior density at time n , $F_{z_n}(\hat{\pi}_{n-1})$, given the posterior from the previous time step, $\hat{\pi}_{n-1}$, and the current observation z_n , is deterministic. However, we cannot exactly reproduce the true posterior $F_{z_n}(\hat{\pi}_{n-1})$, and hence, we have the approximation $\hat{F}_{z_n}(\hat{\pi}_{n-1})$, which itself is the result of a random sampling experiment as done by the PGM algorithm. Therefore, we require that the approximation be a good enough representation of the true posterior with a high probability, where the

probability now is over the randomness of the sampling that is required to generate the approximation.

Lemma 1 *Let $\|\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})\| \leq \epsilon$, for all n . Under Assumption 2, it follows that $\|\hat{\pi}_n - \pi_n\| \leq \frac{(C+1)\epsilon}{1-\rho}$.*

PROOF. We have:

$$\begin{aligned} \hat{\pi}_n - \pi_n &= \hat{F}_{z_n}(\hat{F}_{z_{n-1}}(\dots(\hat{F}_{z_1}(\pi_0))\dots)) \\ &\quad - F_{z_n}(F_{z_{n-1}}(\dots(F_{z_1}(\pi_0))\dots)), \\ &= \underbrace{[\hat{F}_n(\hat{F}_{n-1}(\dots(\hat{F}_1(\pi_0))\dots)) - F_n(\hat{F}_{n-1}(\dots(\hat{F}_1(\pi_0))\dots))]}_{\Delta_n} \\ &\quad + \underbrace{[F_n(\hat{F}_{n-1}(\dots(\hat{F}_1(\pi_0))\dots)) - F_n(F_{n-1}(\hat{F}_{n-2}(\dots(\hat{F}_1(\pi_0))\dots)))]}_{\Delta_{n-1}} \\ &\quad + \dots + \underbrace{[F_n(F_{n-1}(\dots(\hat{F}_1(\pi_0))\dots)) - F_n(\dots(F_1(\pi_0))\dots)]}_{\Delta_1}. \end{aligned} \quad (26)$$

Note that the different terms on the RHS above are:

$$\begin{aligned} \Delta_n &= \hat{F}_n(\hat{\pi}_{n-1}) - F_n(\hat{\pi}_{n-1}), \\ \Delta_{n-1} &= F_n(\hat{F}_{n-1}(\hat{\pi}_{n-2})) - F_n(F_{n-1}(\hat{\pi}_{n-2})), \dots \\ \Delta_1 &= F_n(\dots(F_2(\hat{F}_1(\pi_0))\dots)) - F_n(\dots(F_2(F_1(\pi_0))\dots)). \end{aligned} \quad (27)$$

Using Assumption 2 and the fact that $\|\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})\| \leq \epsilon$, for all n , it follows that:

$$\|\Delta_i\| \leq C\rho^{n-i}\epsilon,$$

and thus,

$$\|\hat{\pi}_n - \pi_n\| \leq \sum_{i=1}^{n-1} C\rho^{n-i}\epsilon + \epsilon \leq \frac{(C+1)\epsilon}{1-\rho}. \quad (28)$$

The above result also holds for initial conditions in the infinite past, i.e., at $n = -\infty$. In the following, we assume that the initial condition was in the infinite past.

Lemma 2 *Let Assumptions 2 and 3 hold. Given any $\delta, \nu > 0$, there exists an $\bar{N} < \infty$, such that:*

$$Prob(\|\hat{\pi}_n - \pi_n\| > \frac{(1+\nu)(1+C)\epsilon}{1-\rho}) \leq \bar{N}\delta, \quad (29)$$

where $\bar{N} = n - n'$, and n' is such that $\sum_{i=-\infty}^{n'} C\rho^{n-i} \leq q$, and $q = \frac{\nu(C+1)\epsilon}{M(1-\rho)}$.

PROOF. Let $e_n = \|\hat{\pi}_n - \pi_n\|$, and let $\epsilon_k = \|\hat{F}_{z_k}(\hat{\pi}_{k-1}) - F_{z_k}(\pi_{k-1})\|$. It follows that $e_n \leq$

$\sum_{k=-\infty}^n C\rho^{n-k}\epsilon_k$. Choose n' such that $\sum_{i=-\infty}^{n'} C\rho^{n-i} \leq q$, where $q = \frac{\nu(C+1)\epsilon}{M(1-\rho)}$. Then:

$$e_n = \underbrace{\sum_{k=n'}^n C\rho^{n-k}\epsilon_k}_{\bar{e}_n} + \underbrace{\sum_{k=-\infty}^{n'} C\rho^{n-k}\epsilon_k}_{\Delta_n^*}. \quad (30)$$

From Assumption 3, it follows that $Prob(\|\Delta_n^*\| > qM) = 0$, and thus,

$$Prob(\|\Delta_n\| > \frac{\nu(C+1)\epsilon}{1-\rho}) = 0. \quad (31)$$

Similarly, from Lemma 1, it follows that:

$$Prob(\bar{e}_n > \frac{(C+1)\epsilon}{1-\rho}) \leq (n - n')\delta \equiv \bar{N}\delta. \quad (32)$$

From (31) and (32), it follows that $Prob(e_n > \frac{(1+\nu)(C+1)\epsilon}{1-\rho}) \leq \bar{N}\delta$.

The two results above establish that if the sampling error at each step in the filter is small enough, and under the condition of exponential forgetting of initial conditions, the true filter density can be approximated arbitrarily closely with arbitrary high confidence. In the following, we establish that the sampling error at each step in the PGM filtering process can be arbitrarily small and thus, it follows from the two results above that the PGM filter can approximate the true filter density with arbitrarily high accuracy and arbitrarily high confidence. First, based on Assumption 1, we define the following:

$$P(\hat{\pi}_{n-1}) \equiv \hat{\pi}_n^- = \sum_{i=1}^{M^-(n)} \hat{\omega}_i^-(n) \mathcal{G}(x; \hat{\mu}_i^-(n), \hat{P}_i^-(n)), \quad (33)$$

$$\hat{P}(\hat{\pi}_{n-1}) \equiv \hat{\pi}_n^+ = \sum_{i=1}^{M^-(n)} \hat{\omega}_i^-(n) \mathcal{G}(x; \hat{\mu}_i^-(n), \hat{P}_i^-(n)), \quad (34)$$

$$F_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\omega}_i(n) \mathcal{G}(x; \hat{\mu}_i(n), \hat{P}_i(n)), \quad (35)$$

$$\hat{F}_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\omega}_i(n) \mathcal{G}(x; \hat{\mu}_i(n), \hat{P}_i(n)). \quad (36)$$

The above represent the true and the approximate PGM predicted and filtered densities at time n given the approximate density $\hat{\pi}_{n-1}$ at time $n - 1$. We have the following result:

Lemma 3 *Given the GMM representation of the prior*

pdf above, and a perfect Clustering algorithm $\mathcal{C}$, given any $\epsilon' > 0$, and $\delta' > 0$, there exists an $N_{\epsilon', \delta'}(n) < \infty$ such that: if the number of samples used to approximate the predicted pdf at time n is greater than $N_{\epsilon', \delta'}(n)$ then:

$$\text{Prob}(|\hat{\omega}_i^-(n) - \omega_i^-(n)| > \epsilon') < \delta', \quad (37)$$

$$\text{Prob}(\|\hat{\mu}_i^-(n) - \mu_i^-(n)\|_2 > \epsilon') < \delta', \quad (38)$$

$$\text{Prob}(\|\hat{P}_i^-(n) - P_i^-(n)\| > \epsilon') < \delta', \quad (39)$$

for all i .

PROOF. Given a random variable (r.v) X , and a function $g(X)$ such that $E(g(X)) = \bar{g}$, and $\text{Var}(g(X)) = \sigma_g^2$, let $\hat{g} = \frac{1}{N} \sum_{i=1}^N g(X_i)$, where X_i are samples of the r.v. X . For large enough N , it follows from the Central Limit Theorem [18] that:

$$\text{Prob}(|\hat{g} - \bar{g}| > \epsilon) < 1 - \int_{-\epsilon}^{\epsilon} \frac{\sqrt{N}}{\sqrt{2\pi}\sigma_g} e^{-\frac{x^2 N}{2\sigma_g^2}} dx. \quad (40)$$

Under the assumption of a perfect clustering scheme, the mixture weights $\hat{\omega}_i^-(n)$ may be assumed to form the event probabilities of a multinomial random variable. It is then straightforward from the central limit theorem (CLT) to find an $N_{\epsilon', \delta'}(\omega_i)$ such that (37) is satisfied. Note that the perfect clustering algorithm is an idealized assumption. In practice, the clustering algorithms may wrongly assign a point to a cluster.

For a Gaussian random vector $X \in \mathbf{R}^n$ with independent components $\{x_1, \dots, x_n\}$ and $E\{x_i^2\} = 1$, it can be shown that

$$P\left\{\|X\|_2 - \sqrt{d} \geq t\right\} \leq 2e^{-\frac{ct^2}{k^4}}, \forall t \geq 0. \quad (41)$$

Here c is a positive constant and $k = \max_i \|x_i\|_{\psi_2}$ where $\|x_i\|_{\psi_2} = \inf\{t; E\{e^{\frac{x_i^2}{t^2}}\} \leq 2\}$ [33]. If $\hat{\mu}_i^-(n)$ is determined as the sample mean of points assigned to cluster C_i , then from (41) we have

$$P\left\{\left\|\left(\frac{\hat{P}_i^-(n)}{N_j}\right)^{-\frac{1}{2}} (\hat{\mu}_i^-(n) - \mu_i^-(n))\right\|_2 - \sqrt{d} \geq t\right\} \leq 2e^{-\frac{ct^2}{k^4}}, \quad (42)$$

where N_j is the number of points assigned to cluster j . This can be manipulated to show that

$$P\left\{\left\|\hat{\mu}_i^-(n) - \mu_i^-(n)\right\|_2 \geq \frac{t + \sqrt{d}}{\left\|\left(\frac{\hat{P}_i^-(n)}{N_j}\right)^{-\frac{1}{2}}\right\|_2 \sqrt{N_j}}\right\} \leq 2e^{-\frac{ct^2}{k^4}}, \quad (43)$$

Using (43) and by setting

$$\frac{t + \sqrt{d}}{\left\|\left(\frac{\hat{P}_i^-(n)}{N_j}\right)^{-\frac{1}{2}}\right\|_2 \sqrt{N_j}} < \epsilon', \quad (44)$$

$$2e^{-\frac{ct^2}{k^4}} < \delta', \quad (45)$$

it is possible to choose an N_j so that (38) is satisfied. The minimum value of N_j that satisfies the above set of equations is chosen as $N_{\epsilon', \delta'}(\mu_j)$.

Let $\hat{P}_i^-(n)$ be the sample average estimate of i^{th} modal covariance. Then it can be shown that

$$P\left\{\frac{\left\|\hat{P}_i^-(n) - P_i^-(n)\right\|}{\left\|\hat{P}_i^-(n)\right\|} \geq ck^2 \left(\sqrt{\frac{d+u}{N_j}} + \frac{d+u}{N_j}\right)\right\} \leq 2e^{-u}, \quad (46)$$

where k is a constant greater than or equal to one and $c \geq 0$ [33]. Here $\|\cdot\|$ represents the spectral norm of the covariance matrix. Hence, by picking an N_j such that

$$ck^2 \left(\sqrt{\frac{d+u}{N_j}} + \frac{d+u}{N_j}\right) \left\|\hat{P}_i^-(n)\right\| < \epsilon', \quad (47)$$

$$2e^{-u} < \delta', \quad (48)$$

the condition given in (39) can be satisfied. Let $N_{\epsilon', \delta'}(C_j)$ be the minimum N_j that satisfies this condition. Pick $N_{\epsilon', \delta'}(j) = \max(N_{\epsilon', \delta'}(\omega_j), \frac{N_{\epsilon', \delta'}(\mu_j)}{\hat{\omega}_j}, \frac{N_{\epsilon', \delta'}(P_j)}{\hat{\omega}_j})$. Such $N_{\epsilon', \delta'}(j)$ can be found for all clusters C_j and given that we choose $N_{\epsilon', \delta'}$ as:

$$N_{\epsilon', \delta'} = \max_j N_{\epsilon', \delta'}(j), \quad (49)$$

it is guaranteed that all the elements of the mean vector $\hat{\mu}^-(n)$ and the covariance matrix $\hat{P}^-(n)$ can be estimated to an accuracy of ϵ' with confidence of at least $1 - \delta'$, which completes the proof of the result.

It may be shown that under Assumption 1 that the Kalman update is an arbitrarily accurate approximation of the true update, the error incurred in estimating the posterior mean and covariance $\hat{\mu}_i(n), \hat{P}_i(n)$ is at most $K(n)\epsilon'$, for some time varying $K(n) < \infty$ which depends on the posterior mean and covariance, given that the predicted prior means and covariances of the clusters of the GMM have been approximated to an accuracy of ϵ' . This can be summarized in the following result:

Lemma 4 Given any $\epsilon', \delta' > 0$, choose $N_{\epsilon', \delta'}(n)$ according (49). If the number of samples used in the PGM filter

to approximate the predicted prior pdf at time n is greater than $N_{\epsilon', \delta'}(n)$ then, there exists $k(n) < \infty$ s.t:

$$\text{Prob}(|\hat{\omega}_i(n) - \hat{\omega}_i(n)| > K(n)\epsilon') < \delta', \quad (50)$$

$$\text{Prob}(\|\hat{\mu}_i(n) - \hat{\mu}_i(n)\| > K(n)\epsilon') < \delta', \quad (51)$$

$$\text{Prob}(\|\hat{P}_i(n) - \hat{P}_i(n)\| > K(n)\epsilon') < \delta', \quad (52)$$

for all i .

Next, we find a bound on the L_1 error between the estimated and true filtered densities given the error between the parameters of the GMM representing the true and the approximate filtered densities.

Lemma 5 Let $|\hat{\omega}_i(n) - \hat{\omega}_i(n)| < \epsilon'$, $\|\hat{\mu}_i(n) - \hat{\mu}_i(n)\| < \epsilon'$, and $\|\hat{P}_i(n) - \hat{P}_i(n)\| < \epsilon$ for all i . Then, given that the state of the system $x \in \mathbb{R}^d$, there exists $C(n) < \infty$ such that $\|\hat{\pi}_n - \hat{\pi}_n\| < C(n)d\epsilon'$.

PROOF. We show the result for the case of a simple one component Gaussian with an error in the covariance, it can be generalized to the GMM in a relatively straightforward fashion but at the expense of a very tedious derivation which we forego here for clarity. We also suppress the explicit dependence on time n in the following for notational convenience.

$$\begin{aligned} \hat{\pi}(x) - \hat{\pi}(x) &= \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} - \\ &\quad \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)}, \\ &\approx \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} \\ &\quad \times \frac{1}{2}(x-\mu)^T (\hat{P}^{-1} \Delta \hat{P}^{-1})(x-\mu), \end{aligned} \quad (53)$$

where $\hat{P} = \hat{P} + \Delta$ and since

$$\begin{aligned} &e^{-\frac{1}{2}(x-\mu)^T (\hat{P} + \Delta)^{-1}(x-\mu)} \\ &\approx e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1} \Delta \hat{P}^{-1}(x-\mu)}, \end{aligned} \quad (54)$$

which in turn implies (53). This in turn implies that:

$$\begin{aligned} \|\hat{\pi} - \hat{\pi}\| &\approx \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} \times \\ &\int e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} \frac{1}{2}(x-\mu)^T \hat{P}^{-1} \Delta \hat{P}^{-1}(x-\mu) dx, \\ &\leq \frac{C(\hat{P})\epsilon'}{(2\pi)^{d/2}|\hat{P}|^{1/2}} \\ &\times \int e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} \frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu) dx, \end{aligned} \quad (55)$$

since there exists $C(\hat{P}) < \infty$ such that $\frac{1}{2}(x-\mu)^T \hat{P}^{-1} \Delta \hat{P}^{-1}(x-\mu) \leq C(\hat{P})\epsilon' \frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)$, owing to the fact that $\|\Delta\| < \epsilon'$.

Now, let $Y = \hat{P}^{-1/2}(X - \mu)$. Then, it follows that:

$$\|\hat{\pi} - \hat{\pi}\| \leq C(\hat{P})\epsilon' \frac{1}{(2\pi)^{d/2}} \int e^{-1/2 y^T y} y^T y dy = C(\hat{P})\epsilon' d. \quad (56)$$

The last step in the above equation follows from noting that $Y^T Y$ is a chi-squared random variable of degree of freedom d and thus, its expected value is d . This establishes our result. In general for a GMM, the constant $C(n)$ would depend on the means and covariances of all the GMM components and their weights.

Lemma 4 and 5 immediately lead us to the following corollary.

Corollary 1 Let $\epsilon'(n)$ be the desired accuracy in estimating the parameters of the GMM representing $F_{z_n}(\hat{\pi}_{n-1})$, i.e., the true filtered density given observation z_n and the PGM posterior pdf at the previous time $\hat{\pi}_{n-1}$. Let $\delta'(n)$ be the desired confidence of the estimate. If $\epsilon'(n)$ and $\delta'(n)$ are chosen such that:

$$C(n)K(n)\epsilon'(n)d = \epsilon, \quad (57)$$

$$\delta'(n) = \frac{\delta}{N}, \quad (58)$$

and the corresponding number of samples $N_{\epsilon'(n), \delta'(n)}(n)$ be chosen according to (49), then it follows that $\|\text{Prob}|\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})| > \epsilon\| \leq \frac{\delta}{N}$.

PROOF. Recall that $\hat{\pi}_n = F_{z_n}(\hat{\pi}_{n-1})$, and $\hat{\pi}_n = \hat{F}_{z_n}(\hat{\pi}_{n-1})$. Then, from Lemma 5 we have that $\|\hat{\pi}_n - \hat{\pi}_n\| \leq C(n)K(n)d\epsilon'(n)$ if $|\hat{\theta}_i(n) - \hat{\theta}_i(n)| < \epsilon'(n)$ for all i , where $\hat{\theta}_i(n)$ represents the true parameters underlying the GMM representation of $\hat{\pi}_n$ and $\hat{\theta}_i(n)$

represents their PGM approximation. Hence:

$$\text{Prob}(\|\hat{\pi}_n - \hat{\pi}_n\| > C(n)K(n)d\epsilon'(n)) < \delta'(n), \quad (59)$$

which owing to the definition of $\epsilon'(n)$ and $\delta'(n)$ leads us to the desired result.

Hence, using Corollary 1 and Lemma 2, it follows that if the number of samples used to approximate the parameters of the predicted GMM pdf at time n is greater than the $N_{\epsilon'(n), \delta'(n)}$, then it follows that $\text{Prob}(\|\hat{\pi}_n - \hat{\pi}_n\| > \frac{(1+\nu)C\epsilon}{1-\rho}) \leq \delta$, for all n for any arbitrarily small $\epsilon, \delta, \nu > 0$. However, in order for Assumption 3 to be valid, the sample averages $\hat{\theta}_n$ have to be almost surely bounded. Here $\hat{\theta}_n$ represents any parameter that is computed from the sample and used to specify the pdf such as the component weights, means or covariances. To show this, due to the Strong Law of Large Numbers, it is also true that $\hat{\theta}_n^N \rightarrow \hat{\theta}_n$ as $N \rightarrow \infty$, where $\hat{\theta}_n^N$ represent the estimate of the parameters after N samples. Given the sample size is large enough, the estimate $\hat{\theta}_n^N$ is arbitrarily close to the true parameters $\hat{\theta}_n$ almost surely, and thus, since the true parameters are bounded, so are the estimates. This may be summarized in the following result.

Proposition 1 *Let Assumptions 1 and 2 hold. Given a perfect clustering algorithm $\mathcal{C}$, and any $\epsilon, \delta, \nu > 0$, at every time step n , choose the required accuracy of the approximation $\epsilon'(n)$ from (57), the required confidence $\delta'(n)$ from (58), and the corresponding minimum number of samples $N_{\epsilon'(n), \delta'(n)}$ from (49), then:*

$$\text{Prob}(\|\hat{\pi}_n - \hat{\pi}_n\| > \frac{(1+\nu)C\epsilon}{1-\rho}) \leq \delta. \quad (60)$$

Several remarks are in order here.

Remark 1 *The above result establishes the convergence in probability of the approximate PGM filter density to the true filtered density uniformly over all time under the assumptions of exponential forgetting of the initial conditions and the adequacy of the Kalman update to approximate the true Bayesian update in the filtering equations. In the absence of the exponential forgetting condition, the convergence result can be obtained only for a finite number of time steps, the development being almost identical. In the absence of Assumption 1, the adequacy of the Kalman update, the result still holds except that there is a new error incurred in sampling the posterior pdf, which will be covered in the companion paper.*

Remark 2 *The analysis above shows that the number of samples required at any time step to ensure the accuracy*

of the filter depends on the current predicted and posterior pdfs, and thus, in general, have to be time varying..

Remark 3 The Curse of Dimensionality: *The number of samples required to estimate the mixture weights does not depend on the dimension of the state space. Additionally, (44), (47) indicate that the number of samples required to estimate the component means and covariances increase only as $O(d)$. From the above analysis, (49), and Lemma 4 it can be concluded that the number of samples required to estimate the parameters of the predicted and posterior pdfs accurately increases only linearly with the dimension of the state space, and thus, is free from the "Curse of Dimensionality". However, we have to be more careful regarding the functional L_1 error in the PGM density: (57) shows that the accuracy parameter required at every time step is inversely proportional to the dimension of the state space since $\epsilon'(n) = \frac{\epsilon}{C(n)K(n)d}$, and thus, in order to attain the same accuracy in terms of the functional error of the filtered density, (40) shows that the number of samples have to increase as $O(d^2)$ where d is the dimension of the problem. Further, it should also be noted that the computation of the sample averages required by the PGM filter grows as $O(d^2)$.*

Remark 4 Compressive Assumption: *The assumption of a finite Gaussian Mixture, is, in our opinion, a compressive argument. We restrict the set of parameters describing the predicted random variable to a finite number that are estimated using sample averages of the predicted random variable. The variance of these random samples is always bounded because of the finiteness assumption, and thus, the number of samples required to estimate the parameters is independent of the dimension of the problem. In general, if we were to find the moments of a random variable from its samples, we need to find all the moments via their sample averages. The variance of the samples of these higher order moments are, in general, not bounded, thereby requiring an infinite number of samples to estimate the pdf.*

3.2 Relationship to other Nonlinear Filters

In this section, we compare and contrast the PGM filter with other nonlinear filters, in particular the PF, the EnKF [8, 12] and some of the GMF in detail.

The prediction stage of the PF is the same as the PGM except that the PF does not get a GMM from the set of predicted particles, and directly uses the Bayesian update on the individual particles, i.e, weights every particle with its likelihood $p(z_n|X_i)$. In the update step lies the computational trouble inherent to the PF, also known as the "particle depletion problem": as the number of dimensions increase, it gets increasingly hard to sample particles with high likelihood, in fact the number of particles goes up exponentially with the number of dimensions thereby subjecting the PF to the curse

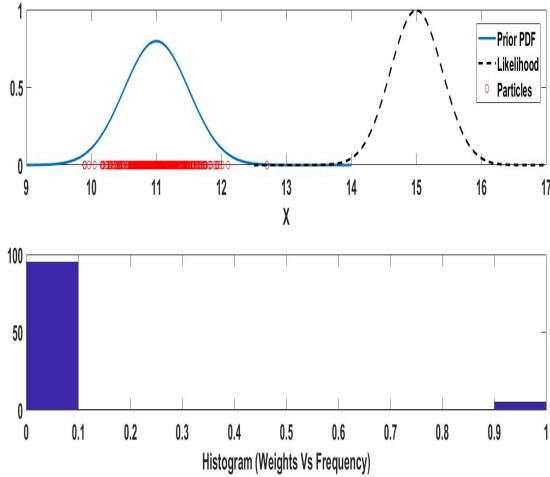


Fig. 2. Particle Depletion

of dimensionality. Consider the simple one dimensional example shown in Fig.2. In this case, a set of 400 particles are sampled from the prior pdf $\pi(x) = \mathcal{G}(x, 11, 0.3)$. The measurement likelihood function is assumed to be $l_z(x) = \mathcal{G}(x, 15, 0.1)$. Since the two pdfs are widely separated, nearly all the weight is allocated to a single particle as observed in the histogram of normalized weights in fig 2. Please see the references [6, 7, 30] for more rigorous insight. In contrast, the PGM uses the Kalman update for the GMM components and thereby does not suffer from the particle depletion problem. Moreover, as shown in the previous section, the number of samples required by the PGM increases only quadratically with the dimension of the problem. In essence, the Kalman update can be thought of as an automatic method to control/ move the predicted particles to the correct regions of the state space given the observation. In fact, this is the philosophy used in the EnKF [12, 14] that perturbs each of the predicted particles using the measurement to obtain a perturbed ensemble of points that actually samples the posterior density. However, the EnKF always assumes a unimodal Gaussian for its predicted and posterior filter densities. At a more minor level, in PGM, we actually do the mean and covariance update of the components using the Kalman update equation rather than perturbing the ensemble of predicted particles [8]. Using the particle based Kalman Filtering method such as the EnKF, we see that at least the calculation of the mean and covariance is independent of the dimension of the state space, and this is precisely the reason why the EnKF is used regularly for the filtering of PDEs such as those arising in Meteorology and Geophysics [3] where even the EKF or UKF are intractable.

The earliest GMFs were a bank of parallel EKFs but the number of modes in the GM was always fixed through both the prediction and the update steps [2]. This, as

has been noted in Section II, can be quite restrictive as it only considers multi-modality arising from initial conditions and never from the prediction and update steps. Other GMFs have more sophisticated methods for updating the weights of the GMM using the Fokker-Planck equation [32] but keeps the number of modes fixed nonetheless. Recent GMF algorithms have also focused on time varying number of modes and used various heuristics to decide when to split a particular Gaussian into multiple components [9]. The problem of covariance splitting during propagation can be avoided given that the mixand covariances are sufficiently small. The blob filter of [26] is a Gaussian mixture filter that tries to circumvent the problem of nonlinearity induced distortion by enforcing an LMI based upper bound on mixand covariances. It uses a novel resampling algorithm to reapproximate the propagated pdf with a GMM so that the bound on the individual mixand covariances is satisfied. It uses EKF for performing the propagation and measurement update steps.

Unlike most of the Gaussian mixture filters discussed here, the PGM filter does not use the typical Kalman filtering propagation methods such as in the EKF/ UKF to propagate and split the PDF. It uses a particle ensemble of the predicted random variable along with a clustering algorithm to conveniently find the number as well as the mean and covariances of the component clusters. In particular, we feel that the PGM harnesses the strength of the PF, the particle prediction step, along with the strength of the Kalman update in GMFs, using clustering algorithms, to develop a technique that is free from the weaknesses of either technique. The GSPF is also a Gaussian mixture filter that relies on particle based propagation. However, there exist significant differences between the prediction and update algorithms of GSPF and the PGM filter. The GSPF relies on an importance sampling based approach to perform the measurement update. Let the predictive distribution be modeled as the weighted Gaussian mixture $\{\omega_i, \mu_i, C_i, i = 1 \dots G\}$. In order to update the i^{th} mixture component, the GSPF samples a set of particles from an importance function $\pi_i(x)$. The particles are then assigned weights based on the ratio between the posterior density and the importance density function. This is similar to the importance sampling based Bayesian update in particle filters. The PGM filter on the other hand updates the components of the predictive density using a Kalman measurement update. The departure from the importance sampling based measurement update as used in particle filters and GSPF to the linearized Kalman update is an important aspect of the PGM filter algorithm. Additionally, let the prior density be modeled as a weighted Gaussian mixture $P_n(X) = \sum_{i=1}^G \omega_i \mathcal{N}(x, \mu_i, C_i)$. Then, for $i = 1 \dots G$, the GSPF samples a set of M points $x_{ni}^j, \{j = 1 \dots M\}$ from the Gaussian component $\mathcal{N}(x, \mu_i, C_i)$. These are then propagated using the dynamic model to obtain

$x_{(n+1)i}^j, \{j = 1 \cdots M\}$. The mean and covariance of the i^{th} component of the predictive distribution is obtained then by evaluating the mean and covariance of $x_{(n+1)i}^j$. The number of mixture components and mixing weights are kept fixed between prior and predictive distributions. Hence the GSPF is essentially a "parallel bank" of Gaussian particle filters. This is not the case with the PGM filters. The PGM filter samples a set of particles directly from the full prior distribution. After propagation, the particles are clustered to obtain the weights, means and covariances that parameterize the mixture model representing the predictive distribution. It also allows the number of components to vary between prior and predictive distributions.

4 Implementation

In this section, we discuss certain steps involved in the practical implementation of the PGM filter in detail.

Modified k-means clustering: The k-means algorithm requires the total number of clusters to be specified externally. To work around this limitation, we have implemented a strategy which only requires the upper bound $M_{max}^-(n+1)$ as the external input instead of $M^-(n+1)$. We define the likelihood agreement measure (L_{mes}) [9] as the measure of fitness of the parametric model θ_a in describing the dataset S . Let $\theta_{a,M}$ be an M -component mixture model indexed by a and arrived at from k -means clustering. Then $L_{mes}(\theta_{a,M})$ may be computed as

$$L_{mes}(\theta_{a,M}) = \sum_{i=1}^{N_p} \pi_{\theta_a}(x_{n+1}^{i-}) \quad (61)$$

where $\pi_{\theta_a}(x)$ is the mixture pdf derived from the parametric model $\theta_{a,M}$. Let θ_{a^*,M^*} be the optimal parametric model with $M_{n+1}^- = M^*$ components that maximizes the L_{mes} given the bound $M_{max}^-(n+1)$. Then, the proposed strategy for clustering is presented in the following algorithm. The most common implementation of the k-means clustering approach is the Lloyd's algorithm [23]. Lloyd's algorithm finds a local minimum to the sum of squares of within-group distances appearing in (23). The time complexity of the Lloyd's algorithm is known to be $O(N_p M d i)$ where N_p is the number of particles to be clustered, d the dimensionality of the state space, M the number of clusters and i is the number of iterations [25]. Implementing the naive clustering strategy as described here will result in a quadratic time complexity in $M_{max}^-(n+1)$. Hence when $M_{max}^-(n+1)$ is too large, it may be suitable to consider any alternative clustering approach that is faster.

Merging: Depending on the clustering scheme, dynamics and measurement models, one may observe several closely distributed mixture modes in the posterior pdf.

Algorithm 2 Clustering Strategy

Input: $S_{n+1}^{-1} = \{x_{n+1}^{1-}, \dots, x_{n+1}^{i-}, \dots, x_{n+1}^{N_p-}\}, M_{max}^-(n+1)$
Output: $\theta_{a^*,M^*}, M^* \leq M_{max}^-(n+1)$
1: $M \leftarrow M_{max}^-(n+1)$
2: $\theta_{a^*,M^*} \leftarrow \theta_{a,M_{max}^-(n+1)}$
3: $L_{mes}^* \leftarrow L_{mes}(\theta_{a,M_{max}^-(n+1)})$
4: **while** $M > 1$ **do**
5: $M \leftarrow M - 1$
6: Compute $\theta_{a,M}$ using k-means
7: **if** $L_{mes}(\theta_{a,M}) \geq L_{mes}^*$ **then**
8: $\theta_{a^*,M^*} \leftarrow \theta_{a,M}$
9: $L_{mes}^* \leftarrow L_{mes}(\theta_{a,M})$
10: **end if**
11: **end while**

The components that are located sufficiently close may be merged to obtain a GMM with well separated modes. A similar situation may arise when the clustering scheme assigns a complex model to describe the data due to overfitting. To identify the right modes to be merged, we define the following normalized error metric [16] as a measure of similarity between modes i and j .

$$D(i, j) = \frac{\int (\mathcal{G}_i(x, \mu_i, P_i) - \mathcal{G}_j(x, \mu_j, P_j))^2 dx}{\int \mathcal{G}_i(x, \mu_i, \Sigma_i)^2 dx + \int \mathcal{G}_j(x, \mu_j, \Sigma_j)^2 dx} \quad (62)$$

Clearly, $D(i, j) = 0$ when the components i, j are identical. It also has an upper bound at 1. Mixture modes that are closely spaced, can be merged whenever the value of normalized error metric falls below a predetermined tolerance (tol). In the present study, we have chosen this tolerance to be $tol = 0.01$. Let $i_1, \dots, i_k$ be the indices of the mixture modes that are to be merged. Then the mixture parameters of the new Gaussian component obtained after merging is given by

$$\begin{aligned} \omega_i &= \sum_{l=1}^k \omega_{i_l}, \quad \mu_i = \frac{\sum_{l=1}^k \omega_{i_l} \mu_{i_l}}{\omega_i} \\ P_i &= \frac{\sum_{l=1}^k \omega_{i_l} (P_{i_l} + (\mu_{i_l} - \mu_i)(\mu_{i_l} - \mu_i)^T)}{\omega_i}. \end{aligned} \quad (63)$$

5 Numerical Examples

In this section, the particle Gaussian mixture filter is applied to three test case problems to evaluate the filtering performance. Due to their ease of implementation and extensive use as benchmark filters, the UKF and PF are included for comparison. The PGM filter is also compared with Gaussian mixture filters such as a GMUKF and the blob filter. Since the propagation and measurement update of a PGM filter is similar to

that of an EnKF, we have also compared it with EnKF in an example problem. For the PF, a sequential importance resampling (SIR) design is considered. The estimation results are assessed for accuracy, consistency and informativeness. A description of the metrics used to compare the filter performance in each of the aforementioned categories is provided below.

1) Accuracy: A Monte Carlo averaged root mean squared error ($E_{rms}(t)$) is considered for evaluating the accuracy of the estimates. The value of $E_{rms}(t)$ is computed over a set of N_{Mo} Monte Carlo runs as

$$E_{rms}(t) = \sqrt{\frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \|\hat{X}^j(t) - \hat{\mu}^j(t)\|_2^2}. \quad (64)$$

Here, $\hat{X}^j(t)$ and $\hat{\mu}^j(t)$ represent the actual and estimated states at the time instant t during the j th Monte Carlo run. The time averaged error ($\overline{E_{rms}}$) can be computed from $E_{rms}(t)$ as

$$\overline{E_{rms}} = \frac{1}{T} \sum_{t=1}^T E_{rms}(t). \quad (65)$$

2) Consistency: The consistency of the filtered pdf is examined using the normalized estimation error squared (NEES) test. For a unimodal state pdf, the NEES test is evaluated using the χ^2 test statistic ($\beta_{j,t}$) given by

$$\beta_{j,t} = (\hat{X}^j(t) - \hat{\mu}^j(t))^T (\hat{P}^j(t))^{-1} (\hat{X}^j(t) - \hat{\mu}^j(t)). \quad (66)$$

The term $\hat{P}^j(t)$ in the above expression represents the covariance of the unimodal filtered pdf at time t during j th Monte Carlo run. The Monte Carlo averaged NEES test (β_t) is computed from this expression as

$$\beta_t = \frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \beta_{j,t}. \quad (67)$$

When the state vector $x \in \mathbb{R}^d$ is normally distributed, the product $N_{Mo}\beta_t$ has a χ^2 distribution with dN_{Mo} degrees of freedom. Hence, the consistency of the filtered pdf can be tested by determining whether β_t falls within probable bounds determined from the corresponding χ^2 random variable.

The NEES test as presented here is not suitable for evaluating the consistency of a multimodal pdf. Let the filtered pdf at time $t = n$ be given by

$$\pi_n(x) = \sum_{i=1}^{M_n} \omega_i \mathcal{G}_i(x; \mu_i, P_i). \quad (68)$$

When the mixture modes are well separated, the total probability that the r.v represented by the GMM belongs to any one of the mixture modes is given by its mixture weight. In defining a GMM describing the state of the dynamical system, the filter hypothesizes the mixture weights, the component means and their covariances. A novel two step procedure for evaluating the consistency of the GMM pdf is developed as part of the present work. Let the random vector V_1 be defined as

$$V_1 = [\mathbb{1}_c^1(x) \cdots \mathbb{1}_c^i(x) \cdots \mathbb{1}_c^{M_n}(x)]^T, \quad (69)$$

where $\mathbb{1}_c^i(x)$ represents the indicator function which equals 1 when the state belongs to the i^{th} mixture component and zero otherwise. Then assuming that the mixture modes are well separated, it can be deduced that

$$E(V_1) = [\omega_1 \cdots \omega_i \cdots \omega_{M_n}]^T. \quad (70)$$

It should be observed that the merging procedure described in the previous section helps to keep the modes well separated as it coalesces the components that are closely spaced. Define

$$\epsilon_v = V_1 - [\omega_1 \cdots \omega_i \cdots \omega_{M_n}]^T, \quad (71)$$

$$\epsilon_v^2 = \epsilon_v^T \epsilon_v. \quad (72)$$

Here the value of V_1 is evaluated over a single instant, i.e, $\sum_{i=1}^{M_n} \mathbb{1}_c^i(x) = 1$. Then, it can be shown that

$$E(\epsilon_v^2) = \sum_{i=1}^{M_n} \omega_i (1 - \omega_i), \quad (73)$$

$$E(\epsilon_v^2 - E(\epsilon_v^2))^2 = \sum_{i=1}^{M_n} \omega_i (1 - \omega_i) ((1 - \omega_i)^3 + \omega_i^3) + \sum_k \sum_{j \neq k} \omega_j \omega_k (\omega_j + \omega_k - 3\omega_j \omega_k) - (E(\epsilon_v^2))^2. \quad (74)$$

Let $V_1^j(t)$ be the above defined indicator vector computed at time t during j^{th} Monte Carlo run. At each instant, the state vector is assumed to belong to the component that maximizes the likelihood given the truth. That is

$$\mathbb{1}_c^i(\hat{X}^j(t)) = \begin{cases} 1, & i = \arg \max \mathcal{G}_i(\hat{X}^j(t), \mu_i, P_i) \\ 0, & \text{otherwise} \end{cases} \quad (75)$$

Let the sum Sw_t be defined as

$$Sw_t = \sum_{i=1}^{N_{Mo}} \frac{\epsilon_v^j(t)^2 - E(\epsilon_v^j(t)^2)}{\sqrt{N_{Mo} E(\epsilon_v^j(t)^2) - E(\epsilon_v^j(t)^2)^2}} \quad (76)$$

The expectations involved in this sum are computed us-

ing the mixture weights $\omega_i^j(t)$ at time t during the run j . Note that when there are two or more mixture modes that are overlapping, they need to be merged first before calculating Sw_t . As N_{Mo} becomes large, the sum Sw_t converges in distribution to a standard Gaussian random variable, $Sw_t \xrightarrow{d} \mathcal{G}(x, 0, 1)$. Hence probability based bounds on the value of Sw_t can be computed from a standard normal distribution. Indeed, the first step in the two step procedure for testing consistency of GMM pdfs involves evaluating Sw_t to determine whether it falls within the desired bounds. In the second step, a NEES test statistic is computed from the GMM except that the mean and covariance of the most likely mode is used to evaluate the χ^2 random variable, i.e.,

$$\beta_{j,t} = (\hat{X}^j(t) - \hat{\mu}_i^j(t))^T (\hat{P}_i^j(t))^{-1} (\hat{X}^j(t) - \hat{\mu}_i^j(t)), \quad (77)$$

$$i = \arg \max N(\hat{X}^j(t), \mu_i^j(t), P_i^j(t))$$

The χ^2 test statistic obtained from the above expression may then be averaged over several Monte Carlo runs to perform the NEES test. This completes the two step consistency check for GMM pdfs.

3) Informativeness: Two separate metrics are considered for evaluating the informativeness of estimates in the present work namely the averaged likelihood of the truth ($L(t)$) and volume of $2 - \sigma$ uncertainty region ($V\sigma_2$). The averaged likelihood of the truth over N_{Mo} Monte Carlo runs may be computed as

$$L(t) = \frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \pi_t^j(\hat{X}^j(t)). \quad (78)$$

Here π_t^j represent the conditional state pdf at time t in the j th Monte Carlo run. The time averaged likelihood is computed from the above expression as

$$\hat{L} = \frac{1}{T} \sum_{t=1}^T L(t). \quad (79)$$

When the state pdf is in the ensemble form, the likelihood is computed using a unimodal Gaussian pdf characterized by the sample mean and covariance of the collection of states. For a well separated GMM pdf, the value of $V\sigma_2$ can be computed as

$$V^j\sigma_2(t) = \sum_{i=1}^{M_t} |2\Sigma_i|, \quad (80)$$

where M_t is the number of modes. Here $|\cdot|$ represents the determinant of the enclosed square matrix. The expression for the unimodal case can be derived by setting $M_t = 1$ in the above equation. We compute the Monte

Carlo averaged $2 - \sigma$ volume as

$$V\sigma_2(t) = \sum_{j=1}^{N_{Mo}} V^j\sigma_2(t). \quad (81)$$

We compute the corresponding time averaged value $\hat{V}\sigma_2$ as

$$\hat{V}\sigma_2 = \frac{1}{T} \sum_{i=1}^T V\sigma_2(t). \quad (82)$$

5.1 Example 1

In this problem, we consider the estimation of the one dimensional discrete time nonlinear dynamic system given by

$$x_{k+1} = \frac{x_k}{2} + \frac{25x_k}{(1+x_k^2)} + 8\cos(1.2k) + \nu_k \quad (83)$$

A measurement model aiding the estimation of the system is specified as

$$z_k = \frac{x_k^2}{20} + n_k, \quad (84)$$

The process noise term ν_k and measurement noise term n_k are assumed to be independent zero mean Gaussian random variables with covariances $Q = 10$ and $R = 1$ respectively. This example or its variants have been studied in several publications before [5, 15]. Two variants of the PGM filter, i.e, PGM1-UT and PGM1, an SIR filter, blob filter and a UKF are simulated to estimate the test case 1 system for a duration of 52 time steps over 50 Monte Carlo runs. The initial state of the system is assumed to be distributed as $P_0(x) = \mathcal{N}(0, 2)$. Measurements are recorded at every other instant. The SIR and the PGM filters are implemented with a set of 50 particles. The upper bound on the number of mixture components M_{max} is set to be 3. For blob filter, 50 Gaussian components were used with a covariance upper bound $P_{max} = 10^{-6}$. The parameter values used in the implementation of the UKF may be found in Table 1.

The values of $E_{rms}(t)$ plotted in Fig.3a indicate good tracking performance by the PGM filters. The PGM filters, the PF and the blob filter are seen to offer comparable tracking performance. The Monte Carlo averaged NEES results plotted in Fig.3b show that the UKF and PF frequently oversteps the upper bound which marks consistent estimates. Furthermore, β_t computed using

Table 1

UKF parameters		
α	β	λ
1.3	1.5	0.2

the PF estimates are found to frequently exhibit peaks several orders of magnitude larger than the 99% upper bound, indicating covariance collapse. For the PGM filters, the component weights are also tested for consistency. For the PGM1-UT, the value of Sw_t is found to stay within the 99 percent bounds during 80.38% of the simulated time. For PGM1, this number was found to be 73%. The averaged likelihood $L(t)$ and the volume $V_{2\sigma}$ plotted in Fig.3c and 3d show that only the blob Filter provides more informative estimates than the PGM-1 and PGM-1 UT.

Table 2

Example 1: Results				
	$\overline{E}_{rms}$	$\beta_{t,c}(\%)$	$\hat{L}$	$\hat{V}_{2\sigma}$
PGM1-UT	6.4513	78.85	0.1209	62.1753
PGM1	6.2859	84.62	0.1253	60.3453
PF	6.5488	46.15	0.1063	79.8123
UKF	8.3279	36.54	0.0488	103.8405
Blob Filter	6.5243	46.15	0.1827	0.0001

The time averaged values of RMSE $\overline{E}_{rms}$, likelihood $\hat{L}$, and the 2σ volume for each filter are listed in table 2. Also included is the fraction ($\beta_{c\%}$) of the time instants during which the computed averaged NEES result stayed within the 99 percent limits,i.e.,

$$\beta_{c\%} = \frac{\sum_{t=1}^T \mathbb{1}_{Ub0.99}(\beta_t)}{T} \quad (85)$$

where $\mathbb{1}_{Ub0.99}(\beta_t)$ is the indicator function which equals 1 when $\beta_t < Ub_{0.99}$ and zero otherwise. The results presented in Table 2 clearly show that the PGM filter implementations offer accurate, consistent and informative estimates.

5.2 Example 2

In this example, the PGM filters are employed in the estimation of a 3 dimensional Lorenz 63 model for atmospheric convection. The noise perturbed dynamics of the Lorenz 63 system is described the the following set of equations,

$$\begin{aligned} \dot{x}_1 &= \alpha(-x_1 + x_2), \quad \alpha = 10 \\ \dot{x}_2 &= \beta x_1 - x_2 - x_1 x_3, \quad \beta = 28 \\ \dot{x}_3 &= -\gamma x_3 + x_1 x_2 + \Gamma(t), \quad \gamma = 8/3 \end{aligned} \quad (86)$$

A scalar nonlinear measurement model(z_k) is considered which is given by

$$z_k = \sqrt{x_1(t)^2 + x_2(t)^2 + x_3(t)^2} + \nu_k. \quad (87)$$

Note that since the measurement function is nonlinear, the PGM filter will not be asymptotically exact. However

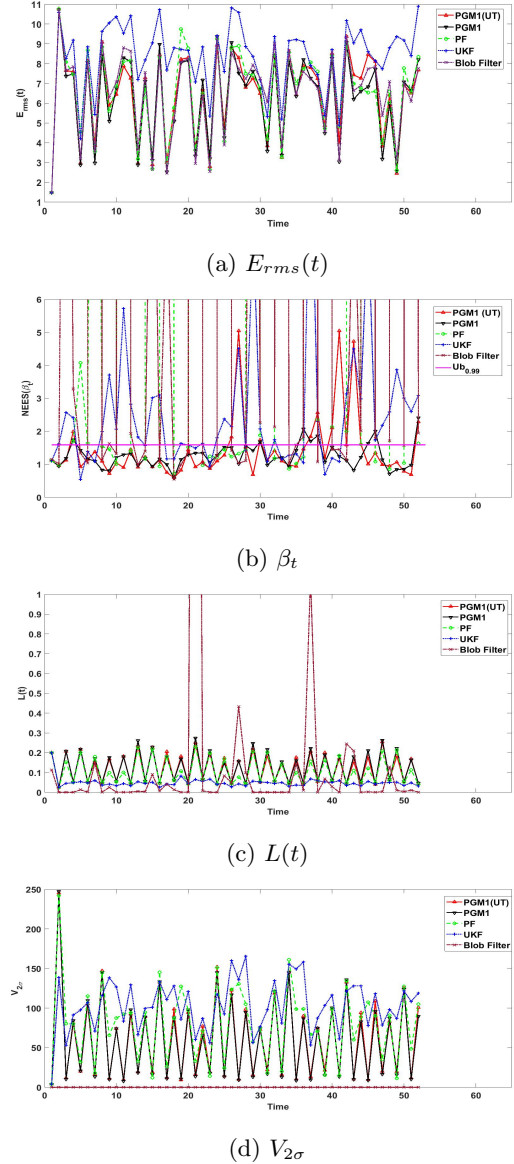


Fig. 3. Example 1 Results

it is still of interest to study the performance of the filter in the nonlinear case. The process and measurement noise covariances are both set be equal to 1. The initial state of the system is characterized by the bimodal pdf

$$p_0(x) = 0.9\mathcal{G}(x, [-0.2, -0.2, 8]^T, \sqrt{0.35}I_{3 \times 3}) + 0.1\mathcal{G}(x, [0.2, 0.2, 8]^T, \sqrt{0.35}I_{3 \times 3}) \quad (88)$$

The state of the system is updated at a time step $\Delta t = 0.01s$. The measurements are recorded at the interval of ten time steps. This example has been considered previously in [32]. The PGM1 filter, PGM1-UT filter, the PF, the blob filter and a conventional Gaussian mixture UKF [2] are employed in the estimation of the Lorenz63 system. The PGM filters and the SIR filter are imple-

mented with 300 particles and M_{max} is set be 3. The UKF is implemented using the parameters listed in table 1. The blob filter is implemented by re-approximating the initial pdf using 300 Gaussians with a maximum covariance $P_{max} = 0.0005 \times I_3$. The values of $E_{rms}(t)$ computed over 50 runs and plotted in Fig.4a show that estimation errors for PGM1 and PGM1-UT are the smallest among the four filters. The Monte Carlo averaged NEES results are plotted in Fig.4b. It is observed that the NEES test statistic β_t for the PF, the blob filter and the mixture UKF overstep the $y = Ub_{0.99}$ line early in the simulation. For the three bi-modal filters PGM1-UT, PGM1 and mixture UKF, the computed value of Sw_t is found to stay within the 99% bounds during 94.12, 96.08 and 89.22 percent of the times considered. The

Fig.4c, 4d show that the PF has the highest average likelihood whereas the Blob filter has the smallest $V\sigma_2$ volume. However, as the NEES results of the PF and the blob filter are seen to stay above the 10^2 for around 85% of the time. The the higher likelihoods and the small $V\sigma_2$ of the PF should be understood as a consequence of its covariance collapse. The consistency fractions($\beta_{c\%}$) and the time averaged values of other performance metrics for each filter are listed in table 3. The results clearly indicate that the PGM filters are more accurate and consistent than the PF, mixture UKF and blob filter.

Table 3

Example 2: Results				
	$\bar{E}_{rms}$	$\beta_{t,c}(\%)$	$\hat{L}$	$\hat{V}\sigma_2(\times 10^4)$
PGM1-UT	14.3886	97.06	0.0038	8.644
PGM1	14.1148	97.06	0.0045	1.046
PF	15.5425	11.76	0.0088	0.6737
GMUKF	15.3528	13.73	0.0019	9.3291
Blob Filter	17.9477	9.80	0.0070	1.09×10^{-11}

5.3 Example 3

In this test case, The PGM filters are employed in the estimation of a Lorenz96 system. The noise perturbed dynamics of the Lorenz96 system is given by

$$\dot{x}_i = x_{i-1}(x_{i+1} - x_{i-2}) - x_i + F + \Gamma(t), \quad (89)$$

where $i = 1, 2, \dots, 40$ [24]. The state variables are assumed to be cyclical so that $x_0 = x_{40}, x_{-1} = x_{39}, x_{41} = x_1$. The term F represents a constant external forcing. In the present work, we set $F = 8$ at which the system is chaotic. The covariance of the zero mean Gaussian white noise is assumed to be $Q = 10^{-2}$. A linear measurement model is employed in the estimation of the Lorenz96 system and it is defined as,

$$z_k = HX_k + \nu_k, \quad H_{i,j} = \begin{cases} 1, & j = 2i - 1 \\ 0, & \text{otherwise.} \end{cases}$$

where $H \in \mathbb{R}_{20 \times 40}$. The measurement noise is assumed to be a zero mean Gaussian random vector with a covariance $R = 10^{-2}I_{20 \times 20}$ where $I_{i,j} = \delta_{i,j}$. The initial state pdf is given by

$$p_0(x) = \mathcal{G}(x, \mu_0, P_0), \quad (90)$$

where $\mu_0 = F [1 \dots 1 \dots 1]^T, \mu_0 \in \mathbb{R}_{40 \times 1}$ and $P_0 = 10^{-3}I_{40 \times 40}$. The state of the system is updated at $\Delta t = 0.05$ time units and measurements are recorded at the interval of 1 time unit. The performance of the PGM-1 filters is compared to that of an EnKF [8], an SIR filter and a blob filter. The PGM-1 filters and the EnKF were equipped with a set of 2000 particles. The value of

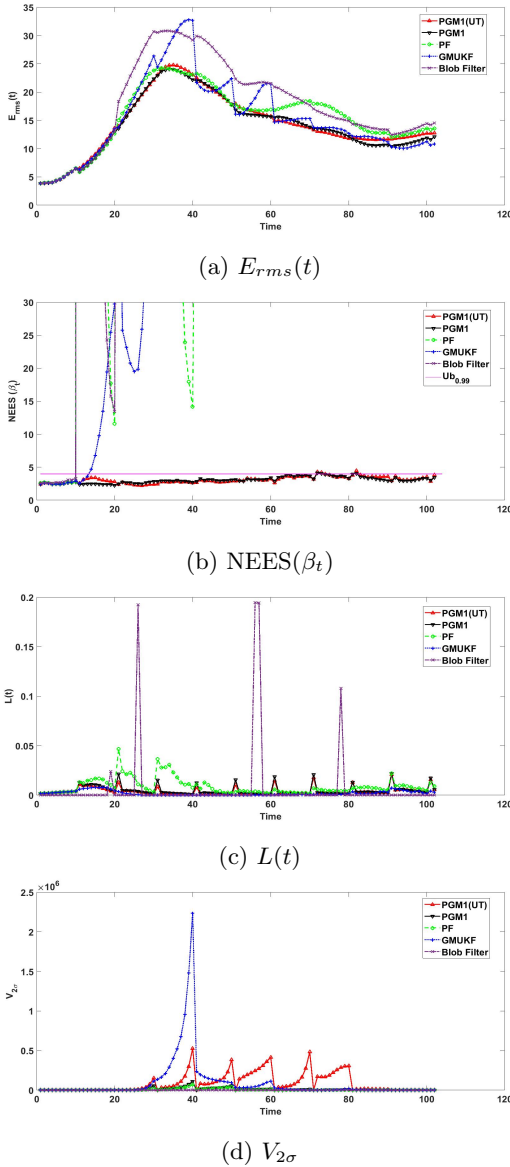


Fig. 4. Example 2 Results

averaged likelihoods($L(t)$) and $V\sigma_2$ volumes plotted in

M_{max} is kept at 2 in order to keep the computational cost low. The SIR filter was implemented with 2000 particles. The blob filter was employed with 2000 Gaussians each having a maximum covariance $P_{max} = 10^{-4} \times I_{40}$. The filters were used to estimate the state of the system for a duration of 200 time steps over 50 Monte Carlo runs. The number of time steps is kept low in order to keep the computational cost of Monte Carlo simulations manageable. The PF and the blob filter were found to undergo covariance collapse after the first measurement was recorded. This was seen to be the case even after using 4000 particles for the PF and using 4000 Gaussians with a maximum covariance of I_{40} for the blob filter. From the $E_{rms}(t)$ plots in Fig. 5a, it can be observed that the tracking performance of the PGM filters and the EnKF are comparable. The Monte Carlo averaged NEES test statistic β_t plotted in Fig. 5b. show that the EnKF and the PGM1 filter offers comparable performance. The value of Sw_t was found to stay within the 99% bounds during 60% of the time for both PGM1-UT and PGM1. The plots of $\log(L(t))$ and $V\sigma_2$ given in Fig. 5c, 5d show that, in comparison to the EnKF, the PGM filters performs better in terms of the $v_{2\sigma}$ volume, whereas the EnKF estimates have the highest averaged likelihoods. The time averaged values of the performance metrics are listed in table 4 along with the consistency fractions. The performances of the EnKF and the PGM filters are seen to be comparable. It may be observed that the EnKF is quite similar to a unimodal PGM Filter as they both rely on particle uncertainty propagation and Kalman measurement update.

Table 4

Example 3: Results				
	$\overline{E_{rms}}$	$\beta_{t,c}(\%)$	$\log \hat{L}$	$\log \hat{V}\sigma_2$
PGM1-UT	18.0069	80.69	89.6553	152.8588
PGM1	18.0452	70.30	89.6227	152.7732
EnKF	18.1055	81.19	89.8193	152.8034

6 Conclusions

A novel Gaussian mixture-particle PGM algorithm for nonlinear filtering has been presented. During the prediction step, the PGM filter uses an ensemble of particles to propagate the prior uncertainty. The propagated ensemble is clustered to recover a GMM representation of the propagated pdf. Measurements are incorporated through a Kalman update of the mixture modes to arrive at the posterior pdf. The PGM approach allows the number and weight of mixture components to be adapted during propagation unlike the conventional mixture filters [2, 31]. Additionally, the PGM is not prone to the curse of dimensionality associated with particle measurement updates. The PGM filter density is shown to converge in probability to the true filter density under the condition of exponential forgetting of initial conditions by the true filter. The PGM filter is employed in

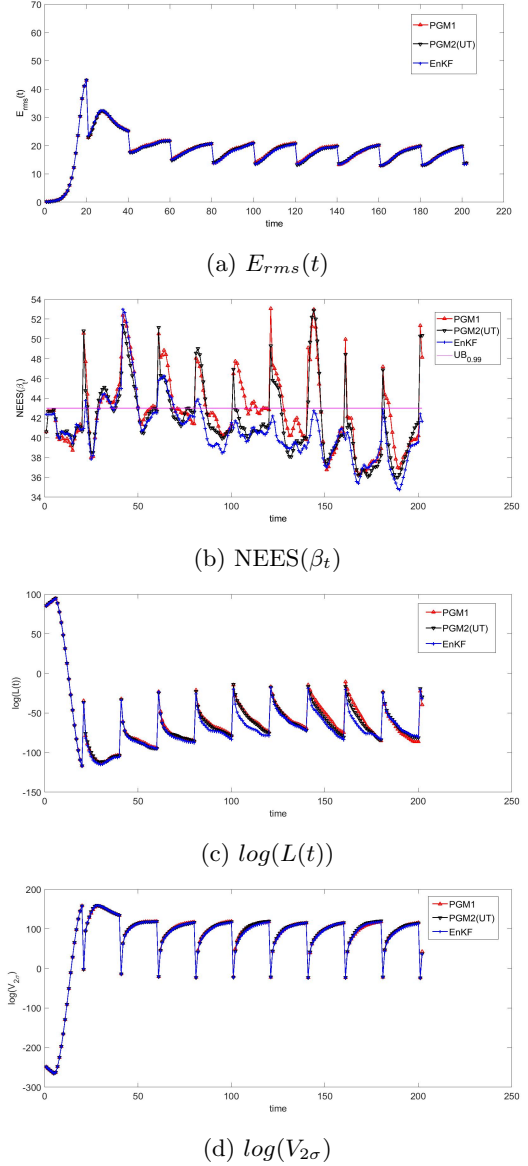


Fig. 5. Example 3 Results

three test cases to evaluate the estimation performance. It is demonstrated that the PGM filter offers superior estimation performance in comparison to UKF, PF the blob filter and a mixture UKF. The PGM filter is demonstrated to be capable of tracking the 40 dimensional lorenz 96 system wherein the PF and blob filter suffers weight depletion. The design of a PGM filtering scheme that incorporates splitting of mixture modes during the measurement update is a goal of future work. Further, the effect of the clustering scheme on the PGM filter needs to be rigorously evaluated.

References

- [1] A.K.Jain, M.N.Murthy, and P.J.Flynn. Data clustering: a review. *ACM Computing Surveys*, 13(3):264–323, 1999.

- [2] D.L. Alspach and H.W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
- [3] Jeffrey L Anderson. Ensemble Kalman filters for large geophysical applications. *IEEE control systems magazine*, 29:66–82, 2009.
- [4] Ienkaran Arasaratnam and Simon Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, 2009.
- [5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2001.
- [6] T. Bengtsson, P. Bickel, and B.Li. Curse-of-dimensionality revisited: Collapse of particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*, volume 2, pages 316–334. 2008.
- [7] Peter Bickel, Bo Li, and Thomas Bengtsson. Sharp failure rates for the bootstrap filter in high dimensions. In *IMS Collections: Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K Ghosh*, volume 3, pages 318–329. 2008.
- [8] G. Burgers, P.J. Van Leeuwen, and G.Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.
- [9] K.J. DeMars, R.H. Bishop, and M.K. Jah. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control and Dynamics*, 36(4):1047–1056, 2013.
- [10] Randal Douc, Eric Moulines, and Ya’Acov Ritov. Forgetting of the initial condition for the filter in general state-space hidden markov chain: a coupling approach. Technical report, 2007.
- [11] Jindrich Dunik, Ondrej Straka, Miroslav Simandl, and Erik Blasch. Random-point-based filters: Analysis and comparison in target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):1403–1421, 2015.
- [12] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, Berlin, 2002.
- [13] Mario.A.T. Figueiredo and Anil.K. Jain. Unsupervised learning of finite mixture models. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.
- [14] G.Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10 143–10 162, 1994.
- [15] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113, 1993.
- [16] Uwe D. Hanebeck, Kai Briechle, and Andreas Rauh. Progressive bayes: a new framework for nonlinear state estimation. In *SPIE vol.5099 Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, pages 256–267, 2003.
- [17] Simon J. Julier, Jeffrey K. Uhlmann, and H.F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, pages 1628–1632, 1995.
- [18] Olav Kallenberg. *Foundations of Modern Probability*. Springer, 1997.
- [19] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of ASME-Journal of Basic Engineering*, 83:96–108, 1961.
- [20] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, (82):35–45, 1960.
- [21] Jayesh H. Kotecha and Petar M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602 – 2612, 2003.
- [22] Tine Lefebvre, Herman Bruynincks, and Joris De Schutter. Comment on ”a new method for the nonlinear transformation of means and covariances in filters and estimators”. *IEEE Transactions on Automatic Control*, 47(8):1406 – 1409, 2002.
- [23] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [24] E.N. Lorenz. Predictability: a problem partly solved. In *Seminar on Predictability, 4-8 September 1995*, volume 1, pages 1–18, Shinfield Park, Reading, 1995. ECMWF, ECMWF.
- [25] Dan Pelleg and Andrew Moore. Accelerating exact k-means algorithms with geometric reasoning. In *KDD ’99 Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–281, 1999.
- [26] Mark L. Psiaki. Gaussian mixture nonlinear filtering with resampling for mixand narrowing. *IEEE Transactions on Signal Processing*, 64(21):10471056, 2016.
- [27] Akkam Veetil Dilshad Raihan and Suman Chakravorty. A ukf-pf based hybrid estimation scheme for space object tracking. In *Proceedings of the AAS/AIAA astrodynamics specialist conference*, pages 4221–4240, August 2015.
- [28] Richard.O.Duda, Peter E.Hart, and David G.Stork. *Pattern Classification*. Wiley-Interscience, New York, 2 edition, November 2000.
- [29] G.L. Smith, S.F. Schmidt, and L.A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical Report NASA TR-135, NASA, January 1962.
- [30] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high- dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.
- [31] H.W. Sorenson and D.L. Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479, 1971.
- [32] G. Terejanu, P. Singla, T. Singh, and P.D. Scott. Adaptive gaussian sum filter for nonlinear bayesian estimation. *IEEE Transactions on Automatic Control*, 56(9):2151–2156, 2011.
- [33] Roman Vershynin. *High Dimensional Probability*. Cambridge University Press, 2017.
- [34] E.A. Wan and R. Van Der Merwe. The unscented Kalman filter. In Simon Haykin, editor, *Kalman Filter and Neural Networks*. J.Wiley and Sons, New York, 2001.
- [35] Tao Yang, Prashant G. Mehta, and Sean P.Meyn. Feedback particle filter. *IEEE Transactions on Automatic Control*, 58(10):2465 –2480, 2013.

A Clustering

Let S be a set of d dimensional vectors, i.e.,

$$S = \{x_1, \dots, x_i, \dots, x_n\}, \quad x_i \in \mathbb{R}^{d \times 1} \quad (\text{A.1})$$

Then, the problem of partitioning S into a group of distinct clusters is called cluster analysis. Clustering involves identifying groups of similar data inside the data set S , grouping them and separating them from the remaining set of vectors. A closely related problem is known as probabilistic clustering or parametric density estimation. The objective of probabilistic clustering is to be able to compute a parametric mixture model describing the distribution of data in S . It is essentially an approach to clustering that is based on probabilistic models wherein data points are assumed to be arriving from distinct mixture modes. In a nutshell, the parametric density estimation problem attempts to find out the model parameters that specify this mixture; e.g. given the data set S , estimate the parameter set $\Theta = \bigcup_{i=1}^L (w_i, \mu_i, C_i)$ describing the Gaussian mixture model that describes the uncertainty in the set S . Several clustering algorithms with widely differing notions of similarity have been proposed. The k-means clustering employed in this paper uses the Euclidean distance in the state space as a metric for similarity.

The Expectation-Maximization (E-M) algorithm is an iterative approach to arriving at a GMM describing a data set using probabilistic methods [28]. The E-M algorithm computes the modal parameters Θ that maximizes the log likelihood of the observed data S . As the name suggests, each iteration of the EM algorithm consists of two steps. In the first step, the expected value of the log likelihood function is computed. In the maximization step, model parameters that maximize this expected log likelihood are found. Both E-M and k-means clustering algorithms can be employed to compute the GMM parameters. However, both of these approaches require the required number of clusters to be specified externally. Clustering approaches such as the Figueiredo-Jain (F-J) algorithm are capable of computing the necessary number of clusters without user supervision [13]. The F-J algorithm computes the optimal number of mixture modes, their weights, means and covariances using an information theoretic criterion. Performance of clustering algorithms can vary significantly between different applications. Hence, the choice of clustering algorithms has to be exercised based on the nature of application at hand, expected accuracy and available computational resources.

Model and Data Based Approaches to the Control of Tensegrity Robots

Ran Wang¹, Raman Goyal¹, Suman Chakravorty¹, and Robert E. Skelton¹

Abstract—This paper proposes two approaches to control the shape of the structure or the position of the end effector for a soft-robotic application. The first approach is a model-based approach where the non-linear dynamics of the tensegrity system is used to regulate position, velocity and acceleration to the specified reference trajectory. The formulation uses state feedback to obtain the solution for the control (tension in the strings) as a *linear programming* problem. The other model-free approach is a novel decoupled data-based control (D2C) which first optimizes a deterministic open-loop trajectory using a black-box (no actual model) simulation model and then develops a linear quadratic regulator around the linearized open-loop trajectory. A two-dimensional tensegrity robotic reacher is used to compare the results for both the approaches for a given cost function. The D2C approach is also used to study two more complex tensegrity examples whose dynamics is difficult to model analytically.

Index Terms—Modeling, Control, and Learning for Soft Robots; Optimization and Optimal Control; Motion and Path Planning

I. INTRODUCTION

THE design of high DOF soft robotic systems has attracted increasing interest in recent years [1], [2]. In this regard, tensegrity structures offer a tantalizing prospect for the principled design of such soft robotic systems. In this paper, we study the modeling and control of high DOF tensegrity robotic systems. In particular, we develop full and reduced order models of class-k tensegrity structures and study both model based and data based approaches to the control of such systems. We also delineate the relative advantages and disadvantages of these approaches.

A. Related Work

Tensegrity structures are designed by placing bars and strings in a methodical arrangement to yield certain desired properties [3]. The dynamics of a tensegrity system can be very accurately modeled as all the members in the system are 1-dimensional elements which only take uni-directional

loading [4]. The absence of bending moments on any individual element not only allows for the accurate modeling of the system but also provides the minimum mass solution to various kinds of loading conditions in engineering mechanics [3], [5]. For the same shape, the stiffness of the structure can also be changed by changing the pre-tension in the strings. The minimal mass architecture along with the variable stiffness characteristic makes it suitable for soft robotic applications like planetary landers [6], flexible robots [7], [8], and deployable space structures [9]. Some of the researchers used model-based approach [7], [8] and some used learning/evolutionary algorithm based approach [6], [10], [11] to control the tensegrity structures but no discussion has been given in the past to compare the two methods.

The control of tensegrity systems amounts to the design of a nonlinear stochastic controller for a very high DOF complex nonlinear system. Controlling an unknown dynamical system adaptively has a rich history in control literature [12] [13]. This classical literature provides rigorous analysis about the asymptotic performance and stability of the closed-loop system, mostly for linear systems or finite state and control space systems. The optimal control of a possibly unknown nonlinear dynamical system with continuous state and action space is a significantly more challenging problem. Even with a known model, computing an optimal control law requires solving a dynamic programming problem. The ‘curse of dimensionality’ associated with dynamic programming makes solving such problems computationally intractable, except under special structural assumptions on the underlying system. *Learning to control* problems where the model of the system is unknown, or is too large or complex for a tractable control synthesis, also suffer from this computational complexity issues, in addition to the usual identifiability problems of adaptive control. For the model-based methods, the computational time is often negligible if analytical model is known.

The past several years have seen significant progress in deep neural networks based reinforcement learning approaches for controlling unknown dynamical systems, with applications in many areas like playing games [14], locomotion [15] and robotic hand manipulation [16]. A number of new algorithms that show promising performance are proposed [17] [18] [19] and various improvements and innovations have been continuously developed. However, despite excellent performance on a number of tasks, reinforcement learning (RL) is still considered very data intensive. The training time for such algorithms is typically really large. Moreover, high variance and reproducibility issues on the performance are also reported [20].

Manuscript received: September 10, 2019; Revised December 11, 2019; Accepted January 26, 2020.

This paper was recommended for publication by Editor Cho, Kyu-Jin upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by NSF NRI award 1637889, NSF CDS&E award 1802867 and AFOSR grant FA9550-17-1-0068 under the Dynamic Data Driven Application Systems(DDAS) program.

¹R. Wang, R. Goyal, S. Chakravorty and R. E. Skelton are with the Department of Aerospace Engineering, Texas A&M University, Texas, USA. {rwang0417, ramaniitrgoyal92, schakrav, bobskelton}@tamu.edu

The codes can be found in our GitHub repository [here](#)
Digital Object Identifier (DOI): see top of this page.

In recent work [21], we proposed a novel decoupled data based control (D2C) algorithm for learning to control an unknown nonlinear dynamical system. Our approach introduced a rigorous decoupling of the open-loop (planning) problem from the closed-loop (feedback control) problem. This decoupling allows us to come up with a highly sample efficient approach to solve the problem in a completely data based fashion. Our approach proceeds in two steps: (i) first, we optimize the nominal open-loop trajectory of the system using a blackbox simulation model, (ii) then we identify the linear system governing perturbations from the nominal trajectory using random input-output perturbation data, and design an LQR controller for this linearized system. We have shown that the performance of D2C algorithm is approximately optimal, in the sense that the decoupled design is near optimal to second order in a suitably defined noise parameter [21]. In this paper, we show the application of the D2C data based control approach to complex and high DOF tensegrity robotic systems, including challenging cases of hard-to-model soft contact constraints and dynamic fluid interactions.

The contributions of the paper are as follows: we show how to model general tensegrity systems and find reduced order models for control design. We present a model based control approach and the application of the data based D2C control approach to such tensegrity systems, and present the relative advantages and disadvantages of the respective methods. We hope the comparison will help to determine the suitable applications for both methods and construct a systematic control solution for tensegrity soft robots.

II. DYNAMICS OF GENERAL TENSEGRITY STRUCTURES

This section provides a brief overview of the non-linear dynamic model of a general tensegrity structure. The final form of class-1 tensegrity dynamics is formulated as a second-order matrix differential equation [4].

$$\ddot{N}M_s + NK_s = W, \quad (1)$$

$$M_s = [C_{nb}^T(C_b^T \hat{J}C_b + C_r^T \hat{m}_b C_r) \quad C_{ns}^T \hat{m}_s], \quad (2)$$

$$K_s = [C_s^T \hat{\gamma} C_{sb} - C_{nb}^T C_b^T \hat{\lambda} C_b \quad C_s^T \hat{\gamma} C_{ss}], \quad (3)$$

where λ represents the force density (compressive force per unit length) in the bar, given by:

$$\hat{\lambda} = -\hat{J}\hat{l}^{-2}[\dot{B}^T \dot{B}] - \frac{1}{2}\hat{l}^{-2}[B^T(W - S\hat{\gamma}C_s)C_{nb}^T C_b^T], \quad (4)$$

and $N = [n_1 \ n_2 \ \cdots \ n_{2\beta+\sigma}] \in \mathbb{R}^{3 \times (2\beta+\sigma)}$ represents the matrix containing the node position vectors $n_i \in \mathbb{R}^{3 \times 1}$, β represents the number of bars, and σ represents the number of string-to-string nodes. The acceleration vector corresponding to the i^{th} node is represented by $\ddot{n}_i$, which forms the matrix $\ddot{N} = [\ddot{n}_1 \ \ddot{n}_2 \ \cdots \ \ddot{n}_{2\beta+\sigma}]$. The bar matrix $B = [b_1 \ b_2 \ \cdots \ b_\beta] \in \mathbb{R}^{3 \times \beta}$ and string matrix $S = [s_1 \ s_2 \ \cdots \ s_\alpha] \in \mathbb{R}^{3 \times \alpha}$ contain bar vectors b_i and string vectors s_i , respectively. The diagonal matrices $\hat{m}_b$ and $\hat{m}_s$ are formed by arranging bar masses and string point masses along the diagonal elements, respectively, and $\hat{J}$ is a diagonal matrix with $J_i = \frac{m_{b_i}}{12} + \frac{m_{b_i} r_{b_i}^2}{l_i^2}$ as the diagonal element that

allows to accommodate the inertia in the bars. The connectivity matrices $C_{nb}, C_b, C_r, C_{ns}, C_s, C_{sb}$, and C_{ss} define the connections between different nodes to form bar vector, string vectors and string-to-string node positions. The external force matrix $W = [w_1 \ w_2 \ \cdots \ w_{2\beta+\sigma}]$ represents the matrix containing external force vector w_i corresponding to each node position vector n_i . The term ‘force density in strings’ is denoted by γ_i to describe tensile force per unit length in the i^{th} string member and $\hat{\gamma}$ represents the diagonal matrix formed from γ_i as its diagonal elements. Equation (4) provides the analytical formula to calculate the diagonal matrix $\hat{\lambda}$ where $[\circ]$ operator sets every off-diagonal element of the square matrix to zero. The diagonal matrix $\hat{l}$ is formed by the length of the bar members where l_i denotes the length of the i^{th} bar. Please refer to [4] for the detailed derivation of this dynamics model.

Dynamics Model for Class-k Tensegrity Structure

The model presented above is formulated for class-1 tensegrity structures. Any class-k structure can be described as a special case of class-1 structures by adding constraints on k nodes to have same node positions. These constraints are written in the linear form as:

$$NP = D, \quad (5)$$

where matrix $P \in \mathbb{R}^{(2\beta+\sigma) \times c}$ and matrix $D \in \mathbb{R}^{3 \times c}$ can be written from the observation to provide constraints and c is the number of added constraints [4]. The added constraints can be accommodated in the class-1 tensegrity dynamics by introducing Lagrange multiplier $\Omega \in \mathbb{R}^{3 \times c}$ that constraints the motion to a certain space by adding Lagrange constraint forces of the form ΩP^T . The full-order dynamics model for class-k tensegrity structure is written as:

$$\ddot{N}M_s + NK_s = W + \Omega P^T, \quad (6)$$

with a modification for force density in the bars as:

$$\hat{\lambda} = -\hat{J}\hat{l}^{-2}[\dot{B}^T \dot{B}] - \frac{1}{2}\hat{l}^{-2}[B^T(W + \Omega P^T - S\hat{\gamma}C_s)C_{nb}^T C_b^T] \quad (7)$$

The degree of freedom for the entire structure has reduced because of the above mentioned constraints. This reduction in the number of allowed displacement dimension can be captured in a reduced dimensional space by reducing the order of the model. The following second order matrix differential equation provides the reduced-order dynamic model as:

$$\ddot{\eta}_2 M_2 + \eta_2 K_2 = \widetilde{W}, \quad (8)$$

where $M_2 = U_2^T M_s U_2$, $K_2 = U_2^T K_s U_2$, $\widetilde{W} = W U_2 - \eta_1 U_1^T K_s U_2$, and the remaining variables are generated from the Singular Value Decomposition (SVD) of the full-column rank matrix P as:

$$NP = NU\Sigma V^T = N[U_1 \ U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} [V^T] = D, \quad (9)$$

where we use $[\eta_1 \ \eta_2] \triangleq NU$ to get:

$$\eta_1 = DV\Sigma_1^{-1}, \quad \dot{\eta}_1 = 0, \quad \ddot{\eta}_1 = 0. \quad (10)$$

Notice that η_1 represents the constraint space in the transformed coordinates and η_2 represents the reduced order space where the motion is present. The Lagrange multiplier is calculated at each step to introduce constraint forces by solving the following algebraic equation:

$$\begin{aligned} \eta_1 U_1^T K_s M_s^{-1} U_1 + \eta_2 U_2^T K_s M_s^{-1} U_1 \\ = W M_s^{-1} U_1 + \Omega V \Sigma_1^T U_1^T M_s^{-1} U_1. \end{aligned} \quad (11)$$

Please refer [4] for the analytic solution of this linear algebra problem.

III. CONTROL FOR CLASS-K TENSEGRITY SYSTEMS

An important step in writing the control of this non-linear dynamic system into a linear programming problem is to be able to write the force densities in the bar $\lambda = [\lambda_1 \lambda_2 \dots \lambda_\beta]^T$ in terms of the linear function of force densities in the strings $\gamma = [\gamma_1 \gamma_2 \dots \gamma_\alpha]^T$. Let us write the i^{th} diagonal element of the matrix $\hat{\lambda}$ from Eq. (7) as:

$$\begin{aligned} \lambda_i = -J_i l_i^{-2} e_i^T [\dot{B}^T \dot{B}] e_i \\ - \frac{1}{2} l_i^{-2} e_i^T [B^T (W + \Omega P^T - S \hat{\gamma} C_s) C_{nb}^T C_b^T] e_i, \end{aligned} \quad (12)$$

which is written using the identity $\hat{x}y = \hat{y}x$, for x and y being the column vectors, and stacking all the scalars into a column vector as:

$$\lambda = \Lambda \gamma + \tau, \quad (13)$$

where

$$\begin{aligned} \Lambda &= [\Lambda_1^T \Lambda_2^T \dots \Lambda_\beta^T]^T, \quad \tau = [\tau_1^T \tau_2^T \dots \tau_\beta^T]^T, \\ \tau_i &= -J_i l_i^{-2} \|\dot{b}_i\|^2 - \frac{1}{2} l_i^{-2} b_i^T (W + \Omega P^T) C_{nb}^T C_b^T e_i, \\ \Lambda_i &= \frac{1}{2} l_i^{-2} b_i^T S (\widehat{C_s C_{nb}^T C_b^T e_i}), \text{ for } i = 1, 2 \dots \beta. \end{aligned}$$

A. Controller for Reduced-Order Dynamics Model

In this section, we write down the control algorithm to control the position of certain nodes in the structure. Let us define the position of those nodes by $Y = LNR$, where L is a matrix that defines the x, y or z coordinates of the node and R matrix defines which nodes to be controlled. $\bar{Y}$ defines the final desired location of the nodes that we want to move from Y to $\bar{Y}$. Therefore, the error in the positions at any time is written as:

$$E = LNR - \bar{Y} = L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}, \quad (14)$$

and the first and second derivatives of the error with respect to time is written as:

$$\dot{E} = L\dot{\eta}_2 U_2^T R, \quad \ddot{E} = L\ddot{\eta}_2 U_2^T R, \quad (15)$$

where $\dot{\eta}_1 = \ddot{\eta}_1 = 0$ was used from the dynamics model formulation. Now, a second-order differential equation in the error dynamics is used to move the nodes from the current

position to the desired position by aptly choosing the control gain parameters matrices Ψ and Θ :

$$\ddot{E} + \dot{E}\Psi + E\Theta = 0, \quad (16)$$

$$L\ddot{\eta}_2 U_2^T R + L\dot{\eta}_2 U_2^T R\Psi + [L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}]\Theta = 0. \quad (17)$$

Further substituting for $\ddot{\eta}_2$ from Eq. (8), the following equation is obtained:

$$\begin{aligned} L(WU_2 - \eta_1 U_1^T K_s U_2 - \eta_2 U_2^T K_s U_2)M_2^{-1} U_2^T R + L\dot{\eta}_2 U_2^T R\Psi \\ + [L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}]\Theta = 0. \end{aligned} \quad (18)$$

Rearranging the above equation to collect all the known and unknown terms together, we get:

$$\begin{aligned} LWU_2 M_2^{-1} U_2^T R + L\dot{\eta}_2 U_2^T R\Psi + [L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}]\Theta \\ = L(\eta_1 U_1^T K_s U_2 + \eta_2 U_2^T K_s U_2)M_2^{-1} U_2^T R. \end{aligned} \quad (19)$$

Let us define the known left side of the equation as:

$$\begin{aligned} C \triangleq LWU_2 M_2^{-1} U_2^T R + L\dot{\eta}_2 U_2^T R\Psi \\ + [L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}]\Theta, \end{aligned} \quad (20)$$

and write the right hand side of the Eq. (19) as:

$$\begin{aligned} L(\eta_1 U_1^T K_s U_2 + \eta_2 U_2^T K_s U_2)M_2^{-1} U_2^T R \\ = LNK_s \underbrace{U_2 M_2^{-1} U_2^T R}_{M_{sn}}. \end{aligned} \quad (21)$$

We now take the i^{th} column of the above matrix and substitute for $K_s = C_s^T \hat{\gamma} C_s - C_{nb}^T C_b^T \hat{\lambda} C_b C_{nb}$ to obtain:

$$\begin{aligned} LNK_s M_{sn} R e_i = LNC_s^T \hat{\gamma} C_s M_{sn} R e_i \\ - LNC_{nb}^T C_b^T \hat{\lambda} C_b C_{nb} M_{sn} R e_i, \end{aligned} \quad (22)$$

and using the identity $\hat{x}y = \hat{y}x$ for the right-hand side terms gives:

$$\begin{aligned} LNK_s M_{sn} R e_i = LNC_s^T (\widehat{C_s M_{sn} R e_i})\gamma \\ - LNC_{nb}^T C_b^T (\widehat{C_b C_{nb} M_{sn} R e_i})\lambda. \end{aligned} \quad (23)$$

Substituting for λ in terms of γ from Eq. (13) gives:

$$\begin{aligned} LNK_s M_{sn} R e_i = -LNC_{nb}^T C_b^T (\widehat{C_b C_{nb} M_{sn} R e_i})\tau \\ + \left(LNC_s^T (\widehat{C_s M_{sn} R e_i}) - LNC_{nb}^T C_b^T (\widehat{C_b C_{nb} M_{sn} R e_i})\Lambda \right) \gamma. \end{aligned}$$

Now, substituting it back to the vector equation of Eq. (19), and stacking up all these matrices on left and vectors on right, we get:

$$[\Gamma_1^T \Gamma_2^T \dots \Gamma_{nr}^T]^T \gamma = [\mu_1^T \mu_2^T \dots \mu_{nr}^T]^T, \quad (24)$$

where

$$\Gamma_i = LNC_s^T (C_s M_{sn} Re_i) - LNC_{nb}^T C_b^T (C_b C_{nb} M_{sn} Re_i) \Lambda,$$

$$\mu_i = Ce_i + LNC_{nb}^T C_b^T (C_b C_{nb} M_{sn} Re_i) \tau,$$

$$C = LWM_{sn}R + L\eta_2 U_2^T R\Psi + [L(\eta_1 U_1^T + \eta_2 U_2^T)R - \bar{Y}]\Theta, \\ M_{sn} = U_2 M_2^{-1} U_2^T, \quad \text{for } i = 1, 2 \dots n_r.$$

B. Controlling the Velocity and Acceleration

For controlling the node positions, we write the error in position as $E_p = L_p N R_p - \bar{Y}_p$ where subscript p is used for the position and write the final linear equation for the force densities in the string in a compact form as (refer Eq. (24)):

$$\Gamma_p \gamma = \mu_p, \quad \gamma \geq 0. \quad (25)$$

For controlling the velocity, we define the error in velocity of certain nodes as:

$$E_v = L_v \eta_2 U_2^T R_v - \bar{Y}_v, \quad \dot{E}_v + E_v \Psi_v = 0, \quad (26)$$

and use a first-order differential equation to derive the error in velocity to zero. Only first derivative of error E_v is required as the control variable, force density γ in the strings come out at the same level of time derivative. Following the same derivation as used in the previous subsections, we write the linear equation to control the nodal velocities as:

$$\Gamma_v \gamma = \mu_v, \quad \gamma \geq 0. \quad (27)$$

To control the acceleration of the nodes, the error is defined as:

$$E_a = L_a \eta_2 U_2^T R_a - \bar{Y}_a, \quad (28)$$

which can be directly converted to a linear equation in control variable by equating it to zero as $E_a = 0$. Following the same procedure, we get the linear algebra equation to solve for control variable as:

$$\Gamma_a \gamma = \mu_a, \quad \gamma \geq 0. \quad (29)$$

Finally, combining the Eqs. (25), (27), and (29) allows to simultaneously control the position, velocity and acceleration of different nodes in the structure.

$$\begin{bmatrix} \Gamma_p^T & \Gamma_v^T & \Gamma_a^T \end{bmatrix}^T \gamma = \begin{bmatrix} \mu_p^T & \mu_v^T & \mu_a^T \end{bmatrix}^T, \quad \gamma \geq 0. \quad (30)$$

IV. DECOUPLED DATA BASED CONTROL (D2C) ALGORITHM

In this section, we propose a data-based approach to solve the tensegrity shape control problem as an alternative to the model-based approach in the previous section. In particular, we apply the so-called decoupled data-based control (D2C) algorithm that we recently proposed [21]. The D2C algorithm is a highly data-efficient Reinforcement Learning (RL) method that has shown to be much superior to state of the art RL algorithms such as the Deep Deterministic Policy Gradient (DDPG) in terms of data efficiency and training stability while

retaining similar or better performance. In the following, we give a very brief introduction to the D2C algorithm (please see [21][22] for the relevant details).

Let us reconsider the dynamics of the tensegrity system given in Eq. (8) and rewrite it in a discrete time, noise perturbed state space form as follows:

$$x_{t+1} = f(x_t) + g(x_t)(u_t + \epsilon w_t), \quad (31)$$

where x_t is the state, u_t is the control, w_t is a white noise perturbation to the system, and ϵ is a small parameter that modulates the noise in the system. Suppose now that we have a finite horizon objective function: $J(x_0) = E[\sum_{t=0}^{T-1} c(x_t, u_t) + \phi(x_T)]$, where $c(x, u)$ denotes a running incremental cost, $\phi(x)$ denotes a terminal cost function and $E[\cdot]$ is an expectation operator taken over the sample paths of the system. The objective of the control design is then to design a feedback policy $u_t(x_t)$ that minimizes the cost function above and is given by:

$$J^*(x_0) = \min_{u_t(\cdot)} E\left[\sum_{t=0}^{T-1} c(x_t, u_t) + \phi(x_T)\right]. \quad (32)$$

The D2C algorithm then proposes a 3 step procedure to approximate the solution to the above problem.

First, a noiseless open-loop optimization problem is solved to find an optimal control sequence, $\bar{u}_t^*$, that generates the nominal state trajectory $\bar{x}_t^*$:

$$J^*(x_0) = \min_{u_t} \left(\sum_{t=0}^{T-1} c(x_t, u_t) + \phi(x_T) \right). \quad (33)$$

subject to the zero noise nominal dynamics: $x_{t+1} = f(x_t) + g(x_t)u_t$.

Second, the perturbed time varying linear system around the nominal given by $\delta x_{t+1} = A_t \delta x_t + B_t \delta u_t$ is identified in terms of the time varying system matrices A_t, B_t .

Third, an LQR controller for the above time varying linear system is designed whose time varying gain is given by K_t . Finally, the control applied to the system is given by $u_t(x_t) = \bar{u}_t^* + K_t \delta x_t$.

Near Optimality of D2C: Let the mean and variance of the true cost function be given by $\bar{J}^*$ and $Var[J]^*$ respectively. Then, it can be shown that the D2C procedure outlined above is near optimal to the second order in the small parameter ϵ in the sense that $|\bar{J} - \bar{J}^*|$ is $O(\epsilon^2)$ and moreover, $\sqrt{Var[J] - Var[J]^*} = O(\epsilon^2)$, where $\bar{J}$ and $Var[J]$ are the mean and variance of the cost corresponding to the D2C policy above (refer to [21] for the details).

The following subsections provide details for each of the above-mentioned steps of the D2C algorithm.

A. Open Loop Trajectory Optimization

A first-order gradient descent based algorithm is proposed here for solving the open-loop optimization problem given in Eq. (33), where the underlying dynamics model is used as a blackbox, and the gradients are found from a sequence of roll-out data with input perturbed by Gaussian noise. Let us denote

the initial guess of the control sequence as $U^{(0)} = \{\bar{u}_t^{(0)}\}_{t=0}^{T-1}$, and the corresponding state trajectory $\mathcal{X}^{(0)} = \{\bar{x}_t^{(0)}\}_{t=0}^T$. The control policy is updated iteratively via:

$$U^{(n+1)} = U^{(n)} - \alpha \nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}, \quad (34)$$

where $U^{(n)} = \{\bar{u}_t^{(n)}\}_{t=0}^{T-1}$ denotes the nominal control sequence after n iterations. $\bar{u}_t^{(n)}$ is a vector of the control value for each control channel at step t and α is the step size parameter. $\bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}$ is the expected episodic cost under $U^{(n)}$ and the corresponding state trajectory $\mathcal{X}^{(n)}$. The gradient vector is defined as:

$$\nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})} = \left(\frac{\partial \bar{J}}{\partial u_1} \quad \frac{\partial \bar{J}}{\partial u_2} \quad \cdots \quad \frac{\partial \bar{J}}{\partial u_T} \right) |_{(\mathcal{X}^{(n)}, U^{(n)})}, \quad (35)$$

which is the gradient of the expected episodic cost w.r.t the control sequence after n iterations. The following paragraph elaborates on how to estimate the above gradient.

Here we define a rollout to be an episode in the simulation that starts from the initial settings to the end of the horizon with a control sequence. For each iteration, multiple rollouts are conducted sequentially, and the gradient vector is updated iteratively after each rollout. The expected episodic cost for the $(n+1)^{\text{th}}$ iteration is calculated by simulating a noiseless rollout with the control sequence $U^{(n)}$. The gradient vector is then estimated in a sequential manner as:

$$\begin{aligned} \nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}^{j+1} &= (1 - \frac{1}{j}) \nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}^j + \\ &\frac{1}{j\sigma_{\delta u}} (J|_{(\mathcal{X}^{j,(n)}, U^{j,(n)})} - \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})})(U^{j,(n)} - U^{(n)}), \end{aligned} \quad (36)$$

where j and n denote the j^{th} rollout of the $(n+1)^{\text{th}}$ iteration. $\bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}$ is the expected episodic cost of the $(n+1)^{\text{th}}$ iteration while $J|_{(\mathcal{X}^{j,(n)}, U^{j,(n)})}$ denotes the cost of the j^{th} rollout under control sequence $U^{j,(n)}$ and the corresponding state trajectory $\mathcal{X}^{j,(n)}$. Note that $U^{j,(n)} = \{\bar{u}_t^{(n)} + \delta u_t^{j,(n)}\}_{t=0}^{T-1}$ where $\{\delta u_t^{j,(n)}\}_{t=0}^{T-1}$ is the zero-mean, i.i.d Gaussian noise added as perturbation to the control sequence $U^{(n)}$. Scalar $\sigma_{\delta u}$ is the variance of the input perturbation and $\nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}^{j+1}$ denotes the gradient vector after j rollouts. The total rollout number r in one iteration is decided by the convergence of the gradient vector. After r rollouts, the control sequence is updated by Eq. (34) in which $\nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}$ is estimated by $\nabla_U \bar{J}|_{(\mathcal{X}^{(n)}, U^{(n)})}^r$ using this iterative method. Repeat this until the cost converges and the optimized nominal control sequence is $\{\bar{u}_t^*\}_{t=0}^{T-1} = \{\bar{u}_t^N\}_{t=0}^{T-1}$. The idea is basically averaging the gradient vector over all the rollouts in an iterative way.

B. Linear Time-Varying System Identification

Closed-loop control design in step 2 of D2C requires the knowledge of the linearized system parameters A_t and B_t for $0 \leq t \leq T-1$. Here we use the standard least square method to estimate these parameters from input-output experiment data.

First start from the perturbed linear system about the nominal trajectory and estimate the system parameters A_t and B_t from: $\delta x_{t+1} = \hat{A}_t \delta x_t + \hat{B}_t \delta u_t$, where $\delta x_t^{(n)}$ is the state perturbation vector and $\delta u_t^{(n)}$ is the control perturbation

vector we feed to the system at step t , n^{th} simulation. All the perturbations are zero-mean, i.i.d, Gaussian noise with covariance matrix σI . σ is a $o(u)$ small value selected by the user. $\delta x_{t+1}^{(n)}$ denotes the deviation of the output state vector from the nominal state after propagating for one step.

Run N simulations for each step and collect the input-output data:

$$Y = [\hat{A}_t \mid \hat{B}_t] X, \quad (37)$$

and write out the components:

$$\begin{aligned} Y &= \begin{bmatrix} \delta x_{t+1}^{(1)} & \delta x_{t+1}^{(2)} & \cdots & \delta x_{t+1}^{(N)} \end{bmatrix}, \\ X &= \begin{bmatrix} \delta x_t^{(1)} & \delta x_t^{(2)} & \cdots & \delta x_t^{(N)} \\ \delta u_t^{(1)} & \delta u_t^{(2)} & \cdots & \delta u_t^{(N)} \end{bmatrix}. \end{aligned} \quad (38)$$

Finally, using the standard least square method, the linearized system parameters are estimated as:

$$[\hat{A}_t \mid \hat{B}_t] = Y X^T (X X^T)^{-1}. \quad (39)$$

C. Closed Loop Control Design

Given the estimated perturbed linear system, we design a finite horizon, discrete time LQR [23] along the trajectory for each time step to minimize the cost function $J = \delta x_T^T Q \delta x_T + \sum_{t=0}^{T-1} (\delta x_t^T Q \delta x_t + u_t^T R u_t + 2\delta x_t^T N u_t)$, subjects to $\delta x_{t+1} = \hat{A}_t \delta x_t + \hat{B}_t \delta u_t$, where $u_t = -K_t \delta x_t$. The feedback gains are calculated as $K_t = (R + B^T P_{t+1} B)^{-1} (B^T P_{t+1} A + N^T)$, where P_t is solved in a back propagation fashion from the Riccati equation: $P_{t-1} = A^T P_t A - (A^T P_t B + N)(R + B^T P_t B)^{-1} (B^T P_t A + N^T) + Q$, $P_T = Q$, $N = 0$. The closed-loop control policy is $u_t(x_t) = \bar{u}_t^* - K_t \delta x_t$, where δx_t is the state deviation vector from the nominal state at time step t .

D. D2C Algorithm: Summary

The Decoupled Data Based Control (D2C) Algorithm is summarized in Algorithm 1.

Algorithm 1: D2C Algorithm

- 1) Solve the deterministic open-loop optimization problem for optimal open-loop control sequence and state trajectory $(\{\bar{u}_t^*\}_{t=0}^{T-1}, \{\bar{x}_t^*\}_{t=0}^T)$ using gradient descent (Section IV-A).
- 2) Linearize and identify the LTV system parameters $(\hat{A}_t, \hat{B}_t)$ via least square (Section IV-B).
- 3) Solve the Riccati equations for each step along the nominal trajectory for feedback gain $\{K_t\}_{t=0}^{T-1}$.
- 4) Apply the closed-loop control policy,

while $t < T$ **do**

$$\begin{aligned} u_t &= \bar{u}_t^* - K_t \delta x_t, \\ x_{t+1} &= f(x_t) + B_t(u_t + \epsilon w_t), \\ \delta x_{t+1} &= x_{t+1} - \bar{x}_{t+1}^* \end{aligned} \quad (40)$$

$t = t + 1$.

end while

V. SIMULATION RESULTS

In this section, we present the application of both the model based control and the data based D2C approach on three tensegrity robotic models with control inputs as the tension in the strings.

A. Structure and Task

We simulated all three robotic models in the MuJoCo simulator [24]: Reacher, Arm and 6-link swimmer. Each of the systems and their tasks are defined as follows:

a) Reacher: The minimal mass solution to compressive loading is provided by T-bar structure [3]. Tensegrity D-bar structure has also been shown to require less mass than a simple continuum bar to take the same load with the added advantage of deployability which makes it suitable for large motion space robotic applications. A T2D1 tensegrity structure is made by combining T-bar and D-bar structures as shown in Fig. 1(a). The two-dimensional structure has 22 bars and 22 strings. The bars in the system are connected by hinge joints reducing the degrees-of-freedom to 14 and the system is controlled by controlling the tension in the 22 control channels (strings). The control task is to move the top tip (end effector) to the target position (red dot).

b) Arm: The arm model is composed of 10 T-bar elements, made from rigidly attaching a vertical bar with a horizontal bar. The horizontal bars are aligned in the direction of the arm length. The first vertical bar is fixed as the base of the model. The model has 18 dimension of states and 38 control channels. The control task is to move the tip to the target position (red dot).

c) 6-link Swimmer: The 6-link swimmer is composed of 6 T-bar elements. The T-bar element here has the same structure as the arm model. Compared with the arm model, the first bar for the swimmer is not fixed so it could swim in the environment. The fluid density for the environment is 3000 kg/m^3 , three times the density of water. The model swims by swaying its body so the horizontal bars would interact with the fluid to generate a forward force. Note that the diameter of the vertical bars is very small (difficult to see in the figure) so that they won't generate too much resistance while swimming. The model has 16 state variables and 22 control channels. The control task is to swim to the target position (red dot).

The initial positions of the models are shown in Fig. 1. The orange objects are the bars, and the grey ones are the strings.

B. Training and Testing

D2C implementation is done in three stages corresponding to those mentioned in the previous section and 'MuJoCo Pro 200 C++ version' is used as the environment for simulating the blackbox model. The model based control (MBC) approach does not require any training time since it is a closed-form solution.

Training (D2C only): The open-loop training plots in Fig. 3 show the cost curves during training. After the cost curves converge, we get the optimal control sequence that could drive the systems to accomplish their tasks. The respective model

positions at the end of the horizon are shown in Fig. 2. The training parameters and outcomes are summarized in (Table I).

Testing Criterion: For the closed-loop design, we proceed with the system identification and feedback gain design step of the D2C algorithm mentioned in the previous section to get the closed-loop control policy. For testing, we compare the performance between the open-loop D2C control policy and the closed-loop D2C control policy under different noise levels. We also compare the D2C closed-loop performance to the MBC design for the Reacher example. The MBC design cannot be applied to the arm or the swimmer because of the moment on the elements and solid-fluid interaction. The open-loop control policy is to apply the optimal control sequence solved in the open-loop training step without any feedback. So the perturbation drives the model off the nominal trajectory and increases the episodic cost as the noise level increases. Zero-mean Gaussian i.i.d. noise is added to every control channel at each step. The standard deviation of the noise is proportional to the maximum control signal in the designed optimal control sequence for D2C. As for MBC, we use the maximum control value in the noiseless nominal control sequence. It must be noted that the range of noise levels (at least up until about 60 % of the maximum nominal control signal) that we are considering here is far beyond what is typically encountered in practical scenarios. As for the criterion for performance, we use episodic cost at each noise level. 500 rollouts are simulated at every noise level tested.

Robustness to Noise: From Fig. 4(a), we can see that both the mean episodic cost and the cost variance of the closed-loop D2C policy is much smaller than that of the open-loop policy for the reacher example throughout the noise level range shown in the plots which prove the success of using D2C on tensegrity models. Although the MBC design can only take up to 30% noise before the simulation fails, it has similar performance with the D2C closed-loop policy when applicable. For the D2C approach, the swimmer example fails after 60% noise, while the reacher and the arm example worked upto 100% noise level. When there is no noise, the analytical solution has the lowest cost of all three. From Fig. 5, it is clear that the control energy used by the MBC solution is way larger than D2C, where the episodic energy is calculated as the L2-norm of the control sequence. This is the result of the optimization-based formulation of D2C. Also, MBC takes up to 30% noise and has higher variance than D2C, which aligns with what is shown in Fig. 4(a). With less control effort, the D2C policy gets a feedback policy close to the accuracy of the analytical solution and endure a wider range of noise. The closed-loop cost for all the three models increases as noise level increases. If we keep increasing the noise level, there exists a threshold that the noise will drive the system too far away from the nominal trajectory that the LQR controller cannot fix it. However, till that point, the closed-loop policy always performs better than the open-loop policy (Fig. 4).

Discussion: The advantage of the MBC approach is that, whenever applicable, it has negligible training time when compared to a data based approach like D2C. However, the D2C approach can achieve the same performance as the model

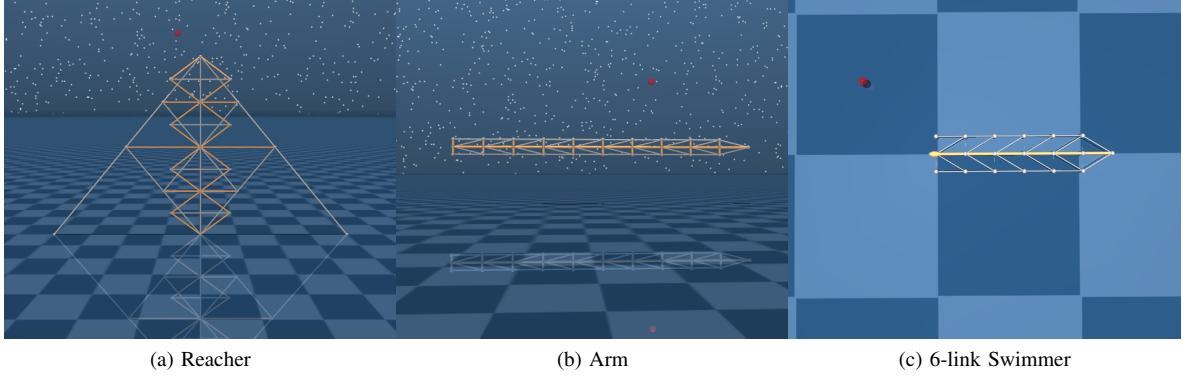


Fig. 1: Models simulated in MuJoCo in their initial states

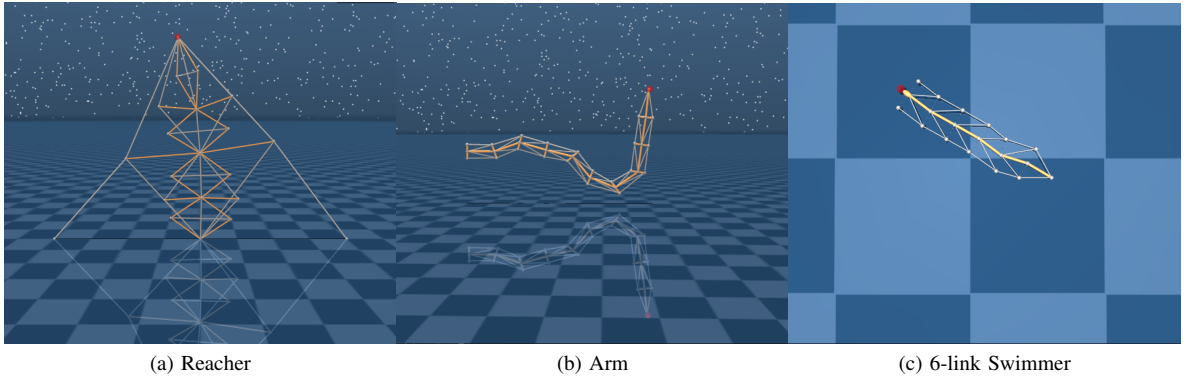


Fig. 2: Models simulated in MuJoCo in their terminal states

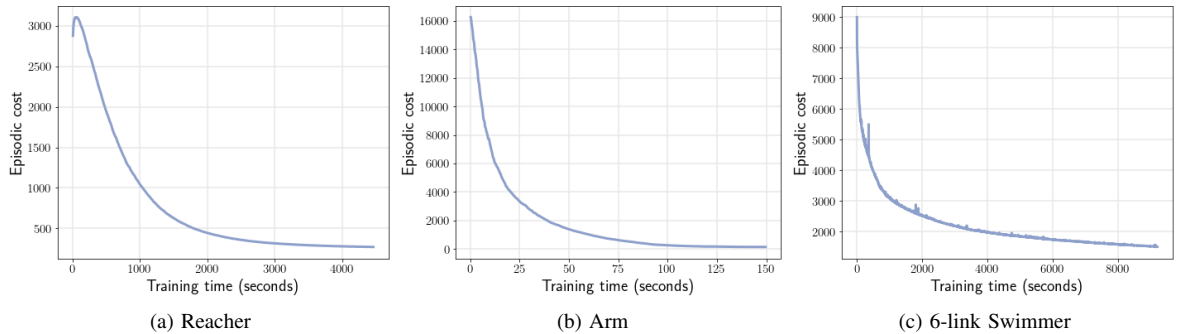


Fig. 3: Episodic cost vs time taken during open-loop training

based approach with far less control effort. One of the reasons for this might be that the reference system needed by the MBC design has not been designed optimally, and thus, a more careful design might result in control efforts that are on par with D2C. In a sense, the philosophies followed by the D2C and the MBC approach are quite similar in that a reference nominal system to be tracked is designed in both, which is then tracked using the closed-loop control. However, the D2C obtains this via optimizing a suitable finite horizon cost of the nonlinear system whereas the MBC approach provides a somewhat arbitrary prescribed behavior, which, in turn, needs

much more control effort in order to be tracked. As mentioned above, the arm and swimmer model are difficult to model due to increased complexity and solid-fluid interaction, but the data based D2C approach can still be applied to these models, indicating a wider application potential for such data based control design approaches.

VI. CONCLUSIONS

In this article, we have provided an overview of the modeling of tensegrity structures that can be used as the mechanism for enabling very high DOF robots, in particular, tensegrity

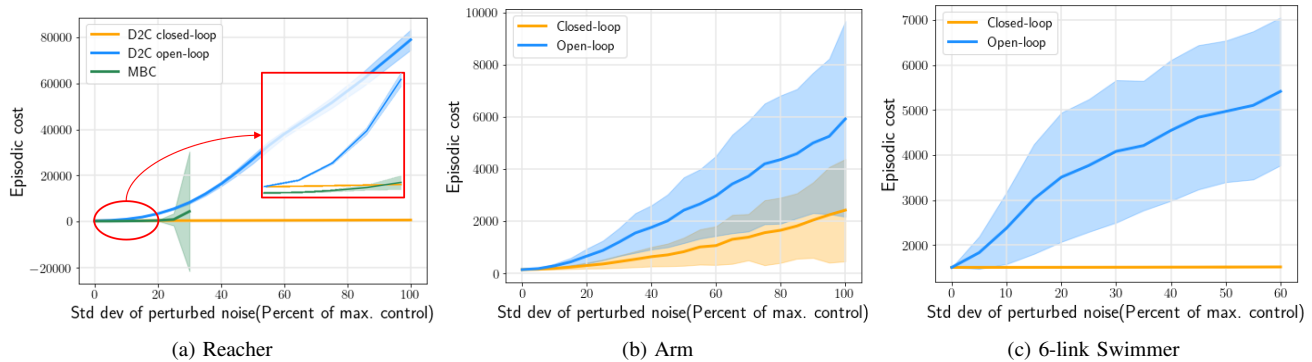


Fig. 4: Performance comparison between D2C open-loop and closed-loop control policy

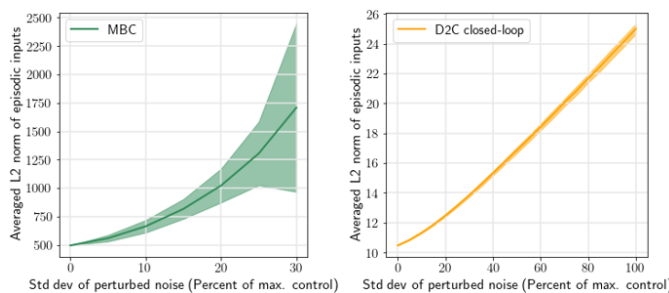


Fig. 5: Control Energy comparison between MBC and D2C

TABLE I: D2C training parameters and outcomes

System	Steps per episode	Time-step (in sec.)	Rollout number	Iteration number	Training time (in sec.)	
					Open-loop	Closed-loop
Reacher	400	0.01	300	1300	4462.1	4.25
Arm	400	0.01	20	600	149.5	4.14
Swimmer	1500	0.006	50	7000	9204.0	5.83

¹ The open-loop training is run on a laptop with I7-7700HQ CPU and 8G RAM. No multi-threading at this point.

robots. We have also studied model-based and data-based approaches to the control of such robots and have outlined their relative merits and demerits. In future work, we shall further develop these control approaches to tensegrity robots, and study the design, construction, and control of such robots. In particular, we plan to apply these techniques to a physical model of tensegrity robotic arm.

REFERENCES

- [1] D. Rus, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 5, pp. 467–475, 2015.
- [2] C. Majidi, "Soft robotics: a perspective — current trends and prospects for the future," *Soft Robotics*, vol. 1, pp. 5–11, 2014.
- [3] R. E. Skelton and M. C. de Oliveira, *Tensegrity Systems*. Springer US, 2009.
- [4] R. Goyal and R. E. Skelton, "Tensegrity system dynamics with rigid bars and massive strings," *Multibody System Dynamics*, vol. 46(3), pp. 203–228, 2019.
- [5] R. E. Skelton and M. C. de Oliveira, "Optimal tensegrity structures in bending: the discrete michell truss," *Journal of the Franklin Institute*, vol. 347(1), pp. 257–283, 2010.
- [6] K. Caluwaerts, J. Despraz, A. İçen, A. P. Sabelhaus, J. Bruce, B. Schrauwen, and V. SunSpiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *Journal of The Royal Society Interface*, vol. 11, no. 98, 2014.
- [7] A. P. Sabelhaus, A. K. Akella, Z. A. Ahmad, and V. SunSpiral, "Model-predictive control of a flexible spine robot," in *2017 American Control Conference (ACC)*, May 2017, pp. 5051–5057.
- [8] H. Karnan, R. Goyal, M. Majji, R. E. Skelton, and P. Singla, "Visual feedback control of tensegrity robotic systems," *2017 IEEE/RSJ-IROS*, pp. 2048–2053, Sept 2017.
- [9] A. G. Tibert and S. Pellegrino, "Deployable tensegrity mast," *In: 44th AIAA/ASME/ASCE, Structures, Structural Dynamics and Materials Conference and Exhibit, Norfolk, VA, USA.*, p. 1978, 2003.
- [10] C. Paul, F. J. Valero-Cuevas, and H. Lipson, "Design and control of tensegrity robots for locomotion," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 944–957, 2006.
- [11] M. Zhang, X. Geng, J. Bruce, K. Caluwaerts, M. Vespignani, V. SunSpiral, P. Abbeel, and S. Levine, "Deep reinforcement learning for tensegrity robot locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 634–641.
- [12] P. R. Kumar and P. Varaiya, *Stochastic systems: Estimation, identification, and adaptive control*. SIAM, 2015, vol. 75.
- [13] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.
- [14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.
- [15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [17] W. Yuhuai, M. Elman, L. Shun, G. Roger, and B. Jimmy, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," *arXiv:1708.05144*, 2017.
- [18] S. John, L. Sergey, M. Philipp, J. Michael I., and A. Pieter, "Trust region policy optimization," *arXiv:1502.05477*, 2017.
- [19] S. John, W. Filip, D. Prafulla, R. Alec, and K. Oleg, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [20] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [21] R. Wang, K. Parunandi, D. Yu, D. Kalathil, and S. Chakravorty, "Decoupled Data Based Approach for Learning to Control Nonlinear Dynamical Systems," *arXiv e-prints*, p. arXiv:1904.08361, Apr 2019.
- [22] D. Yu, M. Rafieisakhaei, and S. Chakravorty, "Stochastic Feedback Control of Systems with Unknown Nonlinear Dynamics," in *56th IEEE Conference on Decision and Control (CDC)*, 2017.
- [23] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Routledge, New York, 1975.
- [24] T. Emanuel, E. Tom, and Y. Tassa, "Mujoco: A physics engine for model-based control," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

An Integrated Approach to Space Situational Awareness

The basic thrust of our research in the project has been in the following major areas, which are listed along with the list of publications resulting from the research. The relevant papers are also attached to this document.

Multi-Target Tracking

1. W. Faber, S. Chakravorty and I. Hussein, "Tracking Space Collisional Cascading via Randomized Finite Set Statistics (FISST)", *The Journal of the Astronautical Sciences*, accepted
2. W. Faber, S. Chakravorty and I. Hussein, "A Belief Space Perspective on RFS based Multi-Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, under review

Data Assimilation

1. D. Raihan and S. Chakravorty, "A Hybrid Particle-UKF for Space Object Tracking", *J. Astronautical Sciences*, 2017, DOI 10.1007/s40295-017-0114-8
2. D. Raihan and S. Chakravorty, "Particle Gaussian Mixture Filters I", *Automatica*, vol. 98, pp. 331-340, 2018
3. D. Raihan and S. Chakravorty, "Particle Gaussian Mixture Filters II", *Automatica*, vol. 98, pp. 341-348, 2018
4. Amirhossein Tamjidi, Suman Chakravorty and Dylan Shell, "Decentralized State Estimation via a Hybrid of Consensus and Covariance Intersection", *Autonomous Robots*, May 2019, <https://doi.org/10.1007/s10514-019-09854-3>
5. A. Tamjidi, R. Otfadeh, S. Chakravorty, D. shell and D. Yu, "Unifying Consensus and Covariance Intersection for Distributed State Estimation over Unreliable Networks", *IEEE Transactions on Robotics*, to appear

The project has also contributed to the development of a Dynamic Data Driven Approach to Planning and Control of unknown systems/ Reinforcement Learning termed D2C.

Dynamic Data Driven Planning and Control/ Reinforcement Learning

1. K. Parunandi and S. Chakravorty, "T-PFC: A Trajectory-Optimized Feedback Control Approach", *IEEE Robotics and Automation Letters*, vol. 4, n. 4, pp. 3457-3463, Oct. 2019
2. R. Wang, R. Goyal, S. Chakravorty and R. E. Skelton, "Model and Data based Approaches to the Control of Tensegrity Robots", *IEEE Robotics and Automation Letters*, vol. 5, n. 3, pp. 3846-3853, 2019, DOI: 10.1109/LRA.2020.2979891

3. R. Wang, K. Parunandi, S. Chakravorty and D. Kalathil, “A Decoupled Data based Approach for Learning to Control Nonlinear Systems”, *IEEE Transactions on Automatic Control*, to appear