PARTICLE GAUSSIAN MIXTURE FILTERS FOR NONLINEAR NON-GAUSSIAN BAYESIAN ESTIMATION

A Dissertation

by

DILSHAD RAIHAN AKKAM VEETTIL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Suman Chakravorty
Committee Members,	John Valasek
	Srinivas Rao Vadali
	Panganamala R. Kumar
Head of Department,	Rodney Bowersox

May 2019

Major Subject: Aerospace Engineering

ABSTRACT

Nonlinear filtering is the problem of estimating the state of a stochastic nonlinear dynamical system using noisy observations. It is well known that the posterior state estimates in nonlinear problems may assume non-Gaussian multimodal probability densities. We present an unscented Kalman-particle hybrid filtering framework for tracking the three dimensional motion of a space object. The hybrid filtering scheme is designed to provide accurate and consistent estimates when measurements are sparse without incurring a large computational cost. It employs an unscented Kalman filter (UKF) for estimation when measurements are available. When the target is outside the field of view (FOV) of the sensor, it updates the state probability density function (PDF) via a sequential Monte Carlo method. The hybrid filter addresses the problem of particle depletion through a suitably designed filter transition scheme. The performance of the hybrid filtering approach is assessed by simulating two test cases of space objects that are assumed to undergo full three dimensional orbital motion.

Having established its performance in the space object tracking problem, we extend the hybrid approach to the general multimodal estimation problem. We propose a particle Gaussian mixture-I (PGM-I) filter for nonlinear estimation that is free of the particle depletion problem inherent to most particle filters. The PGM-I filter employs an ensemble of randomly sampled states for the propagation of state probability density. A Gaussian mixture model (GMM) of the propagated PDF 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 PGM-I filter is capable of handling the non-Gaussianity of the state PDF arising from dynamics, initial conditions or process noise. A more general estimation scheme titled PGM-II filter that can also handle non-Gaussianity related to measurement update is considered next. The PGM-II filter employs a parallel Markov chain Monte Carlo (MCMC) method to sample from the posterior PDF. The PGM-II filter update is asymptotically exact and does not enforce any assumptions on the number of Gaussian modes.

We test the performance of the PGM filters on a number of benchmark filtering problems chosen from recent literature. The PGM filtering performance is compared with that of other general purpose nonlinear filters such as the feedback particle filter and the log homotopy based particle flow filters. The results also indicate that the PGM filters can perform at par with or better than other general purpose nonlinear filters such as the feedback particle filter (FPF) and the log homotopy based particle flow filters. Based on the results, we derive important guidelines on the choice between the PGM-I and PGM-II filters. Furthermore, we conceive an extension of the PGM-I filter, namely the augmented PGM-I filter, for handling the nonlinear/non-Gaussian measurement update without incurring a large computational penalty. A preliminary design for a decentralized PGM-I filter for the distributed estimation problem is also obtained. Finally we conduct a more detailed study on the performance of the parallel MCMC algorithm. It is found that running several parallel Markov chains can lead to significant computational savings in sampling problems that involve multi modal target densities. We also show that the parallel MCMC method can be used to solve global optimization problems.

DEDICATION

To my family

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Professor Suman Chakravorty. His unrelenting support and guidance has been invaluable to me throughout my graduate studies. He has had a tremendous impact in my personal and professional growth. I shall be forever in his debt for the example he set and for his insights about research and career.

I would also like to thank my committee members, professors John Valasek, Srinivas Rao Vadali and Panganamala R Kumar, for their time, patience and insightful feedback. During the course of my research, I had the great pleasure of working with Dr. Islam Hussein and Prof. Dylan Shell. I extend my sincere thanks to them for helping me broaden my knowledge and perspective.

I would like to acknowledge members of Prof. Chakravorty's research group: Weston Faber, Amir Hossein Tamjidi, Reza Oftadeh, Dan Yu, Saurav Agarwal, and Parnandi Karthikeya for their ingenious suggestions and persistent help. I gratefully acknowledge John P. John whose inputs on utilizing high performance research computing has had a tremendous impact on my research. Special thanks to my colleagues Divya Praturi, Ankita Mittal, Dipanjan Saha, Niladri Das, Atanu Halder, Stoian Borissov, Hari Krishnan and Jobin Joy who helped me in countless ways during my graduate studies.

I would like to thank the staff in the Department of Aerospace Engineering, particularly Gail Rowe, Rose Sauser and Karen Knabe for their unfailing help and support. Finally, I would like to thank my friends Deepthi Sen and Benjamin Simington. I cannot begin to describe my gratitude to them for all the love, kindness and encouragement that they have given me during these years.

CONTRIBUTORS AND FUNDING SOURCES

This work was funded by AFOSR grant number: FA9550-13-1-0074, FA9550-17-1-0068 under the Dynamic Data Driven Application Systems (DDDAS) program and the NSF award ECCS163788. This work was supported by my Ph.D advisor Dr. Suman Chakoravorty and my dissertation committee members Dr. John Valasek, Dr. Srinivas Vadali and Dr. Panganamala R. Kumar. Other support provided by my collaborators Dr. Dylan Shell, Dr. Weston Faber and Dr. Islam I. Hussein. All other work conducted for the dissertation was completed by the student independently.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Recent advances in engineering and computer science have made it possible to bring a wide variety of systems, larger in scale and complexity than ever before, within the scope of automation. Supervision and control of a dynamical system or industrial process requires the knowledge of its underlying state. Consequently, there is a growing interest in recursive and computationally efficient algorithms for estimating the state and associated uncertainty in higher dimensional nonlinear systems. A filter is an algorithm that makes use of the information about the time evolution of a system and its relationship with the recorded measurements to provide estimates of its current state. The limitations in exact modeling of physical phenomena and the ubiquity of noise necessitates probabilistic representation of state estimates. In this thesis, we consider the Bayesian approach to estimation, that allows prior knowledge concerning the statistics of the state to be combined with the likelihood of measurement data. We propose a unifying framework for incorporating the nonlinearity of the process and measurement models and non-Gaussian statistics of the noise and state variables within a Bayesian estimation paradigm*. We examine the strengths and limitations of various existing estimation algorithms and propose combining them to secure better estimation performance. The state PDF, in its functional form, is modeled as a GMM to incorporate the effects of non-Gaussianity and multimodality. The exponential growth in computational requirements with the increasing dimension of the state space, also known as the 'Curse of Dimensionality', has been a major challenge to the scalability of several existing estimation algorithms. We attempt to establish non exponential bounds on the required computational resources.

*Parts of this chapter were reprinted with permission from [1, 3, 2, 4].

Additionally, we propose incorporating machine learning algorithms to enhance the filtering performance. We consider it a central objective to obtain rigorous guarantees on estimation performance. Extensive simulation studies are conducted to compare the performance of the proposed filtering approach to existing methods. In section 1.1 we briefly review the two step recursive Bayesian approach to estimation. In section 1.2 we discuss the essentials of mixture model filtering with emphasis on Gaussian mixture models.

1.1 Preliminaries: Bayesian Filtering

Let $x \in \Re^d$ be the state of a dynamical system given by

$$x_{t+1} = f(x_t, w_t), \tag{1.1}$$

where w_t is a noise term with known distribution. Let $z_1, z_2, \cdots, z_n$ be a sequence of measurements of the system where

$$z_k = h(x_k) + \nu_k. \tag{1.2}$$

The distribution of the measurement noise term ν_k is assumed to be known. Given this state space description and the initial state PDF $P(X_0)$, the objective of the filtering problem is to be able to determine the conditional state PDF $P(X_t|Z_t)$. Here Z_t represents the sequence of all measurements recorded until time t.

The transition kernel $p_n(x/x')$ of the state Markov chain can be derived from the process model given in equation 1.1. Given the transition kernel $p_n(x/x')$ and the measurement likelihood $p_n(z_n|x)$, the filtered density of the state Markov chain can be computed using a recursive algorithm that involves two basic steps. Let π_{n-1} be the PDF of the state at time $n-1$ conditioned on Z_{n-1}. Given π_{n-1}, the prediction

step evaluates the propagated prior $\pi_n^-(x)$, i.e., the PDF of the state at n conditioned on Z_{n-1}, using the law of total probability.

$$\pi_n^-(x) = \int p_n(x/x')\pi_{n-1}(x')dx', \tag{1.3}$$

In the measurement update step, the propagated PDF $\pi_n^-(x)$ is updated with the new measurement z_n according to the Bayes rule to obtain the posterior PDF $\pi_n(x)$.

$$\pi_n(x) = \frac{p_n(z_n/x)\pi_n^-(x)}{\int p_n(z_n/x')\pi_n^-(x')dx'}, \tag{1.4}$$

The prediction and the update steps above are the key steps to any recursive filtering algorithm. In practice, obtaining closed form expressions for the integral in equation 1.3 and the integral denominator term in equation 1.4 are extremely difficult except in a few cases. Most nonlinear filters attempt to obtain good approximations to these integrals.

1.2 Preliminaries: Mixture Model Filtering

Let us assume that a mixture representation has been chosen for the predicted and posterior PDFs. In particular, let:

$$\begin{aligned} \pi_n^-(x) &= \sum_{i=1}^{M^-(n)} \omega_i^-(n) p_{i,n}^-(x), \\ \pi_n(x) &= \sum_{i=1}^{M(n)} \omega_i(n) p_{i,n}(x), \end{aligned} \tag{1.5}$$

where $p_i^-(.), p_i(.)$ are standard 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:

$$\pi_n^-(x) = \sum_{i=1}^{M(n-1)} \underbrace{\omega_i(n-1)}_{\omega_i^-(n)} \underbrace{\int p_n(x|x')p_{i,n-1}(x')dx'}_{p_{i,n}^-(x)}. \tag{1.6}$$

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

$$\omega_i^-(n) = \omega_i(n-1), \tag{1.7}$$

$$p_{i,n}^-(x) = \int p_n(x|x')p_{i,n-1}(x')dx'. \tag{1.8}$$

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

$$\pi_n(x) = \frac{\sum_{i=1}^{M^-(n)} \omega_i^-(n)p_n(z_n|x)p_{i,n}^-(x)}{\sum_{i=1}^{M^-(n)} \omega_i^-(n) \int p_n(z_n|x')p_{i,n}^-(x')dx'}. \tag{1.9}$$

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

$$l_i(n) \equiv \int p_n(z_n|x')p_{i,n}^-(x')dx'. \tag{1.10}$$

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

$$\pi_n(x) = \sum_{i=1}^{M^-(n)} \underbrace{\frac{w_i^-(n)l_i(n)}{\sum_j w_j^-(n)l_j(n)}}_{w_i(n)} \underbrace{\frac{p_n(z_n|x)p_{i,n}^-(x)}{l_i(n)}}_{p_{i,n}(x)}. \tag{1.11}$$

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:

$$\omega_i(n) = \frac{w_i^-(n) l_i(n)}{\sum_j w_j^-(n) l_j(n)}, \tag{1.12}$$

$$p_{i,n}(x) = \frac{p_n(z_n|x) p_{i,n}^-(x)}{l_i(n)}. \tag{1.13}$$

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:

$$p_{i,n-1}(x) = \mathcal{G}(x; \mu_i(n-1), P_i(n-1)), \tag{1.14}$$

where $\mathcal{G}(x; \mu, P)$ represents the Gaussian PDF with mean μ and covariance P. Consider first the prediction equations. Note that from equation 1.6, 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

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{x_1}{2} \\ \sin(\frac{x_2}{2}) \end{bmatrix} + \Gamma(t), \tag{1.15}$$

$$\pi_0(X) = \mathcal{G}(X, \begin{bmatrix} -12 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.2 & 0 \\ 0 & 1 \end{bmatrix}), \tag{1.16}$$

where $\Gamma(t)$ is a white noise process.

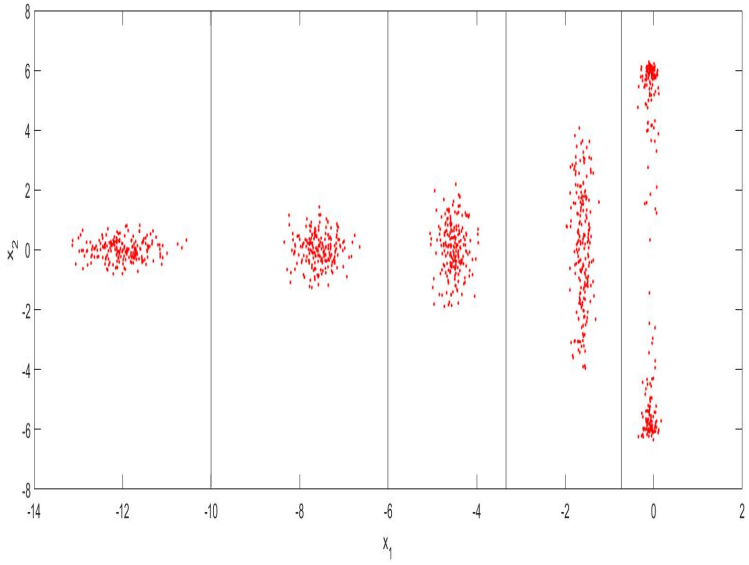


Figure 1.1: Formation of multimodality through dynamics. Reprinted with permission from [1, 2].

In Figure 1.1, 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 equation 1.13. Since the prior component is Gaussian, and if the update equation 1.13 is approximated using the Kalman/ least squares update [8, 5], we have :

$$\mu_i(n) = \mu_i^-(n) + P_{i,zx}^T(n)P_{i,zz}^{-1}(n)(z_n - E_i[h(X)]), \tag{1.17}$$

$$P_i(n) = P_i^-(n) - P_{i,zx}^T(n)P_{i,zz}^{-1}(n)P_{i,zx}(n), \tag{1.18}$$

where,

$$P_{i,zx}(n) = E_i[h(X) - E_i(h(X))(X - E_i(X))^T], \tag{1.19}$$

$$P_{i,zz}(n) = E_i[(h(X) - E_i(h(X))(h(X) - E_i(h(X))^T]. \tag{1.20}$$

Here $E_i[f(X)]$ represents the 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 1.13. An illustration of this is given in Figure 1.2. 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

$$z = x_1^2 + \tau, \ \tau \sim \mathcal{G}(x, 0, 2). \tag{1.21}$$

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. A fully nonlinear filter that employs a Gaussian mixture representation of the state

PDF must be able to split/fuse individual mixture modes when necessary.

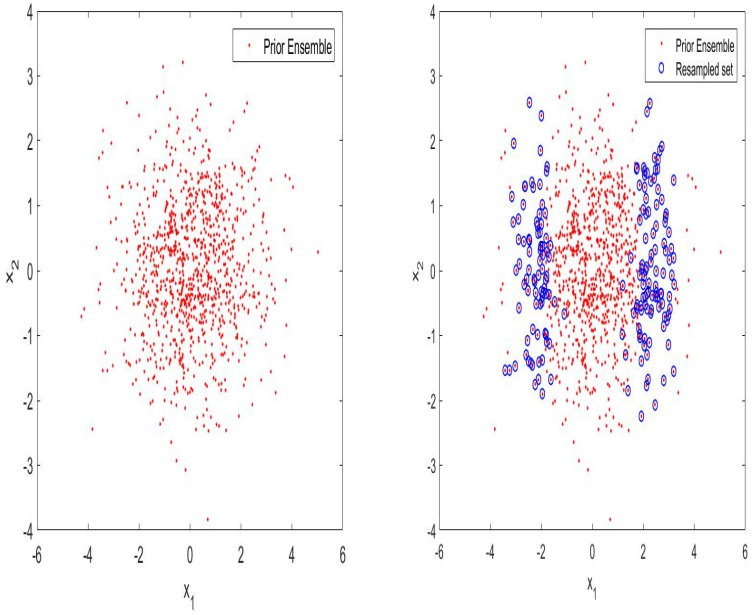


Figure 1.2: Formation of multimodality through measurement update. Reprinted with permission from [1, 3, 2, 4].

1.3 Outline

In the next chapter, we extensively review relevant previous literature and discuss the context of our contributions. In chapter 3, we consider a space object tracking problem and formulate a hybrid filtering approach that is capable of providing accurate and consistent estimates even when the measurements are sparse. In chapter 4, we discuss the PGM-I filter, an extension of the hybrid filtering approach to incorporate multimodal state PDFs arising from nonlinear uncertainty propagation. A generalization of this method, namely the PGM-II filter, that can also handle multimodal measurement updates is discussed in chapter 5. The PGM-II filter relies on a

parallel MCMC method to obtain samples from multimodal posterior distributions. In chapter 6, we test the performance of the PGM filters performance in various benchmark problems. We study how dimensionality, nonlinearity, sample size etc influence the estimation performance. In chapter 7, we develop a preliminary design for a decentralized PGM-I filter for the distributed estimation problem. In chapter 8, we test the performance of the parallel MCMC method, first developed in chapter 5, in sampling and optimization problems. Lastly, we summarize the conclusions of this work and discuss directions for future research in chapter 9. An illustration of the structure of this dissertation is given in Figure 1.3.

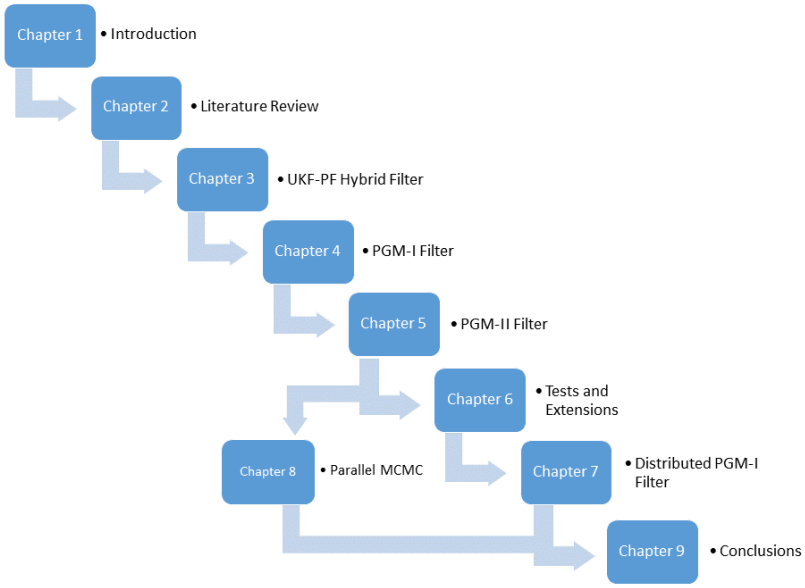


Figure 1.3: Structure of the dissertation.

2. LITERATURE REVIEW*

In this chapter we survey previous work that are relevant to the contributions of this thesis. The optimal linear estimator, known as the Kalman filter, set the framework for recursive estimation of uncertain dynamical systems using state space description [9, 10]. The Kalman filter furnishes the unbiased minimum variance estimates when the dynamical system is linear and the uncertainties involved are Gaussian. The extended Kalman filter (EKF) was introduced to incorporate nonlinear systems into the Kalman filtering framework [11]. However, the limitations of the Jacobian linearization assumptions and the accumulation of linearization errors can result in the divergence of EKF estimates. The emergence of sigma point Kalman filters, specifically the UKF, gave rise to a derivative free alternative to the EKF [12, 13, 14]. The UKF computes the statistics of the state PDF using carefully chosen and weighted sigma points. It was found to consistently outperform the EKF at a comparable computational cost. However, as the UKF approximates the statistics of the posterior PDF using the first two moments, it can be ineffective in the estimation of a general multimodal non-Gaussian PDF. Handling the non-Gaussianity of the state PDF is crucial in problems such as space object tracking wherein the measurements may be sparse and the PDF may undergo extensive distortion induced by nonlinearities. In such cases, the UKF may even produce suboptimal and diverging estimates [15]. Cubature Kalman filters (CKF) that rely on a spherical-radial cubature rule to evaluate the integrals involved in the estimation have been proposed [16]. A variant of cubature filters, that perform random scaling and rotation of cubature points and axes, known as stochastic integration filter (SIF)

*Parts of this chapter were reprinted with permission from [1, 3, 2, 4].

has also been proposed recently [17].

2.1 Gaussian Mixture Filters

A Gaussian mixture approximation of the state PDF was proposed to incorporate the multimodality of the problem in nonlinear settings [18, 19]. 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 [20]. A different approach to improving the accuracy of GMM filters is by splitting the Gaussian components during the propagation based on nonlinearity induced distortion [21]. Both of these approaches 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 [22]. 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.

2.2 Ensemble/ Particle Filters

The particle filters (PF) are a class of sequential Monte Carlo methods that employ an ensemble of states known as particles to represent the state PDF [23, 24]. 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 [25]. 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 [26, 27]. The Gaussian sum particle filter (GSPF) is a nonlinear filter that uses a GMM representation of the state PDF [28]. 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.

3. THE SPACE OBJECT TRACKING PROBLEM*

In this chapter, we consider the probabilistic estimation of a space object. The dynamics model of the orbiting objects and an angles-only measurement model are discussed in the section 3.1. We propose a novel unscented Kalman particle hybrid filtering framework which incorporates select features of both UKF and PF to produce a fast and accurate nonlinear filter that can be employed for space object tracking [5, 6]. A detailed account of the filter design process is provided in section 3.3. The state PDF in the space object tracking problem is assumed to be unimodal. However, the more general PGM-I filtering algorithm discussed in chapter 4 can be obtained as the natural multimodal extension of the hybrid filtering approach proposed here. The hybrid filter is employed in the estimation of two tests cases of LEO objects in the section named Simulations and Results.

3.1 Dynamics & Measurement Models

This section contains a brief description of the perturbed dynamics of orbiting objects. Following this, an angles-only measurement model, employed to aid state estimation, is described.

3.1.1 Dynamics of space objects

The acceleration experienced by an object, of mass m_o in the inverse square gravitational field of earth is given by

$$\mathbf{a_g} = -\frac{GM_e m_o \mathbf{r}}{r^3}. \tag{3.1}$$

*Parts of this chapter were reprinted with permission from [6, 5].

Here G is the universal gravitational constant, $\mathbf{r}$ the vector joining the center of earth to the center of mass (CM) of the object, r its magnitude and M_e the mass of earth. The gravitational acceleration, as given in equation 3.1, assumes that the central body is spherically symmetric. In reality, the earth has a non-symmetric mass distribution similar to an oblate ellipsoid, with more mass distributed along the equator. To account for the non-sphericity, the gravitational potential is expanded into a series of spherical harmonics. The dominant perturbation term in the resulting expansion is called the J_2 harmonic. The perturbing acceleration arising from the J_2 term, $\mathbf{a_{J_2}}$ is given by

$$\mathbf{a_{J_2}} = -\frac{3}{2} J_2 \frac{GM_e}{r^2} \left(\frac{r_{eq}}{r}\right)^2 \begin{bmatrix} (1 - 5(\frac{x_3}{r})^2)\frac{x_1}{r} \\ (1 - 5(\frac{x_3}{r})^2)\frac{x_2}{r} \\ (3 - 5(\frac{x_3}{r})^2)\frac{x_3}{r} \end{bmatrix}, \tag{3.2}$$

where r_{eq} is the equatorial radius of the earth and x_1, x_2, x_3 are the Cartesian coordinates of the CM of the object measured from the center of earth [29]. In addition to this, the orbital motion is also affected by the non-conservative atmospheric drag which may be significant in low earth orbits. Assuming a blunt form factor, the acceleration due to atmospheric drag force can be computed as

$$\mathbf{a_D} = -\left(\frac{A_{cs}}{m_o}\right) \rho C_d \left(\frac{v^2}{2}\right) \mathbf{i_v}. \tag{3.3}$$

Here A_{cs} is the cross sectional area of the object, C_d, the drag coefficient, and ρ the atmospheric density. The term v represents the magnitude of relative velocity between atmosphere and orbiting object whereas $\mathbf{i_v}$ is the unit vector along its direction. A simple exponential model may be employed to describe the variation of

atmospheric density with altitude, according to which

$$\rho(r) = \rho_0 \exp\left(-\left(\frac{r - r_0}{H}\right)\right). \tag{3.4}$$

Here ρ_0 and r_0 are reference density and radius. The variable H, known as scale height, is the vertical distance over which the density of the atmosphere reduces by a factor of mathematical constant e. The resultant acceleration experienced by the object is then given by

$$\ddot{\mathbf{r}} = \mathbf{a_g} + \mathbf{a_{J_2}} + \mathbf{a_D}. \tag{3.5}$$

In this study we have ignored the effects of other forces such as third body perturbations since we consider only space objects in low earth orbits(LEO). The perturbation terms present in the resultant acceleration force the object to undergo full three dimensional motion. Hence the state of the system is taken to be

$$X = [x_1\ x_2\ x_3\ \dot{x_1}\ \dot{x_2}\ \dot{x_3}]^T. \tag{3.6}$$

where x_1, x_2, x_3 are the Cartesian coordinates of the object measured with respect to an inertial frame placed at the center of earth. In practice, we integrate equation 3.5, which describes a continuous time dynamical system, numerically with fixed time steps Δt. This allows us to obtain an approximate discrete time solution $f(X_t)$ which maps the state X_t at time t to that at $t + \Delta t$, i.e., $X_{t+\Delta t}$. In addition to the acceleration terms described in the equation (3.5), a sequence of independent and normally distributed noise terms are added in the state update equations to account for modeling uncertainties.

3.1.2 Measurement Model

Let $\mathbf{r}$ and $\mathbf{r_s}$ be the inertial position vectors of the space object O and the ground station respectively. Then the relative position of the object with respect to the ground station is given by

$$\mathbf{r^i} = \mathbf{r} - \mathbf{r_s}. \tag{3.7}$$

The sensor measures the topocentric inclination (θ) and right ascension (ϕ) from a ground station assumed to be located on the earth's equator. The coordinatization of the relative position vector of the space object with respect to the ground station, in the station frame, may be computed by multiplying the inertial vector $\mathbf{r^i}$ with the appropriate orthonormal transformation matrix. If the effects due to precession, nutation etc. of the earth are neglected, then the ground station is in an elemental rotation about the polar axis with respect to the inertial frame. At $t = 0$, both ground frame and inertial frame are aligned. Assuming a constant spin rate ω for the earth, the transformation matrix for the ground station at time t may be calculated as

$$F(t) = \begin{bmatrix} \cos\omega t & \sin\omega t & 0 \\ -\sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.8}$$

If $[r_x^i \ r_y^i \ r_z^i]$ are the Cartesian coordinates of the object in the ground frame, then θ and ϕ are calculated as

$$\theta = \sin^{-1}\left(\frac{r_z^i}{r^i}\right) \tag{3.9}$$

$$\phi = \tan^{-1}\left(\frac{r_y^i}{r_x^i}\right) \tag{3.10}$$

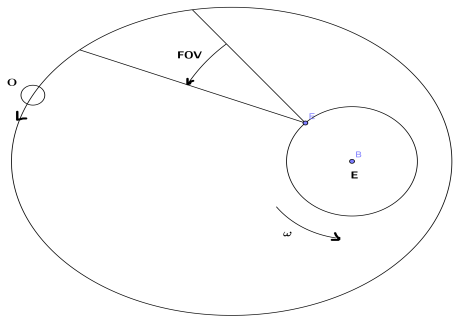


Figure 3.1: The sensor is fixed on the ground station which defines a non inertial frame that spins with the earth. Reprinted with permission from [5].

where $r^i = \sqrt{{r_x^i}^2 + {r_y^i}^2 + {r_z^i}^2}$.

A zero mean Gaussian measurement noise with 3.9 arcsec standard deviation is assumed. The FOV of the ground station is limited by 75 degrees on either side in the azimuthal direction and by 90 degrees on either side in the polar direction. An illustration of the space object-ground station system is presented in Figure 3.1. Once the space object is inside the FOV of the sensor, measurements are registered with a preset probability of detection P_d. It must be noted here that the P_d is a state independent factor employed solely for the purpose of ensuring that the measurements are not recorded at all instants and that they come in at random even when the object is within the FOV.

3.2 Assumptions

Before proceeding further we state two key assumptions:

Assumption 1. *We shall assume that the filtered state PDF of the object while inside the FOV can be well approximated by a unimodal PDF.*

This may not be a bad assumption given that frequent disambiguating measurements are recorded when the space object is within the FOV of the sensor.

The orbital perturbations that influence the dynamics of space objects are well known. Consequently, accurate computational models that can simulate the time evolution of space objects are available. Employing an accurate model helps to minimize the variations due to differences between modeled and actual dynamics. We also assume that high precision models that limit the statistical variability in the time evolution of the space object are available. Based on these observations we make the following assumption.

Assumption 2. *We assume that the magnitude of process noise affecting the dynamics of the space object is minimal. The process noise terms are sampled as acceleration terms drawn from a zero mean Gaussian PDF with covariance* $10^{-18} I_3$.

3.3 The Unscented Kalman-Particle Hybrid Filter

In this section we discuss the UKF-PF hybrid filtering approach to space object tracking. The hybrid filtering approach presented here is formulated on the basic premise that a carefully chosen combination of multiple approximation schemes may accord a higher overall estimation performance than any one particular scheme, owing to the high specificity of individual schemes' performance to the estimation scenario under study. In particular, the UKF-PF hybrid filter proposes to employ a particle approximation of the state PDF when the space object is outside the FOV of sensors and to switch to a unimodal approximation when the object is inside the observation range. The approximate unimodal PDF is characterized by a mean and covariance. The rationale behind this particular selection of approximation methods is discussed below.

When the space object is outside the observation range, the state PDF undergoes extensive nonlinearity induced distortion. In the absence of measurements, particles from the PDF at time $t+1$, i.e., $p(X_{t+1}|Z_t)$, may be obtained by merely propagating

the ensemble at time t through the dynamic model as this is equivalent to choosing $q(X_{t+1}|X_t) = p(X_{t+1}|X_t)$. This is a key advantage of the particle approximation as the Dirac delta kernels are allowed to freely evolve with time without enforcing any restrictive assumptions on the nature of state PDF. Moreover, by increasing the number of particles, the approximation may by refined to any degree of accuracy. Additionally, particle based methods do not require any auxiliary optimization or entropy calculations to incorporate the effects of distortions. This makes the particle approach well suited for keeping track of the evolution of state PDF when the object is outside the observation range. Moreover, as no measurement update is performed outside the FOV, the sample weights remain constant. Hence, the particle based estimator can be employed outside the FOV without facing the prospect of particle depletion and the associated curse of dimensionality. As a result, a tracking scheme that employs a particle approximation only during the flight of the object outside the FOV can use a much smaller number of particles in comparison to a full PF implementation.

Availability of frequent measurements during the flight of the object inside the FOV ensures that the growth and distortion of uncertainty between consecutive measurements will be limited. The hybrid filter employs a unimodal Gaussian density to approximate the state PDF during this stage. The expectation integrals involved in the calculation of mean and covariance of the propagated state can be evaluated using the unscented transform (UT). These are then used to fit a Gaussian PDF for the propagated PDF. The measurement update step in a standard UKF is free from resampling procedures that are customary to sequential Monte Carlo methods. As the propagated PDF and measurement random variable are assumed to be Gaussian, the UKF employs a Kalman measurement update to compute the mean and covariance of the posterior PDF. The UKF measurement update resembles the linear minimum

variance update that appears in the Kalman Filter. The Kalman gain K_k that is used to update the estimate is computed as $K_k = C_{XZ}C_{ZZ}^{-1}$. The covariance C_{ZZ} and cross covariance C_{XZ} can also be computed using UT. Additionally, the standard UT uses only $2n + 1$ sample points for evaluating the integrals in an n-dimensional estimation problem. Hence it is computationally efficient. The sequencing of filters that underlies the UKF-PF hybrid estimator is illustrated in Figure 3.2.

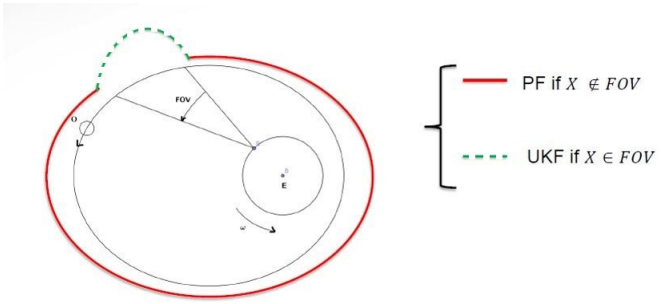


Figure 3.2: The hybrid approach attempts to harness the merits of both PF and UKF. Reprinted with permission from [6].

In practice, when the estimated position of the object exits the FOV of the sensor, we sample a set of equally weighted particles from the unimodal Gaussian state PDF. Subsequently, they are propagated forward in time while maintaining the individual particle weights $w(X^i)$ constant till the reentry of the object into FOV. The hybrid filter switches from the particle set to the unimodal UKF at the re-entry of the object into the sensor's FOV, which is marked by the reappearance of measurements. During this transition, a unimodal density has to be retrieved from the ensemble of particles while incorporating the additional information obtained

from the newly recorded measurement. During the operation of the hybrid filter, a measurement update is performed on the particles solely at this PF-UKF transition stage. It needs to be emphasized that as the state space is six dimensional and observations are sparse, the estimator my face a significant risk of particle depletion at this stage. The new measurement information may be incorporated into the state PDF in a number of ways, for e.g. by performing a direct particle measurement update on the ensemble [24]. Alternatively, the posterior mean and covariance may be determined by refining the prior mean and covariance through a Kalman update step [9]. In this study, we have considered three approaches for transitioning from PF to UKF. In hybrid filter 1, the PF-UKF transition is accomplished through a particle measurement update. In hybrid filters 2 and 3, the posterior statistics are computed via a Kalman update. However, the filters 2 and 3 adopt different approaches to computing the Kalman gain. The three transition designs are discussed in detail below.

Hybrid Filter 1

Once the object re-enters the FOV and the first measurement is registered, all N particles are assigned weights based on their respective likelihoods derived from the

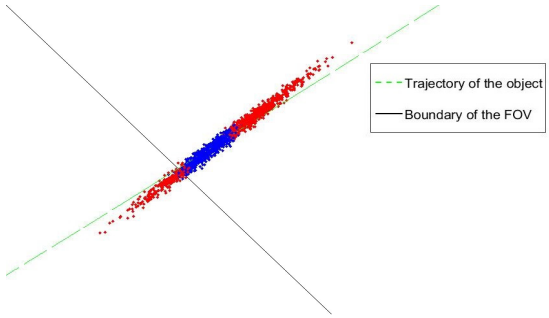


Figure 3.3: Ensemble of particles before (red) and after (blue) resampling. Reprinted with permission from [5].

measurement model. The weight assigned to each particle $X^i_{t_r}$ at any instant t_r may be computed as

$$w(X^i_{t_r}) = \frac{p(z_{t_r}|X^i_{t_r})}{\sum_{j=1}^{N} p(z_{t_r}|X^j_{t_r})}, \tag{3.11}$$

where z_{t_r} is the measurement recorded at that instant. The mean (μ_{X,t_r}) and covariance (C_{X,t_r}) of this weighted sample may be computed as

$$\mu_{X,t_r} = \sum_{i=1}^{N} w(X^i_{t_r}) X^i_{t_r}, \tag{3.12}$$

$$C_{X,t_r} = \sum_{i=1}^{N} \frac{w(X^i_{t_r})(X^i_{t_r} - \mu_{X_{t_r}})(X^i_{t_r} - \mu_{X_{t_r}})^T}{1 - \Sigma_{j=1}^{N} w(X^j_{t_r})^2}. \tag{3.13}$$

The $\mu_{X_{t_r}}$ and $C_{X_{t_r}}$ described in equations (3.12) and (3.13) are then used to initialize the approximate unimodal PDF required for the subsequent UKF based estimation.

Figure 3.3 shows the particle distribution in the x-y plane during the transition from PF to UKF. Once the importance weights are updated with measurement likelihoods, the contribution from several particles to the posterior PDF $p(X_{t_r}|Z_{t_r})$ diminishes due to their getting negligible weights. The disparity in weights is revealed when the particle ensemble is resampled. Particles with negligible weights are discarded during resampling. The updated state distribution composed of the retained particles is given in Figure 3.3. Prior to resampling the propagated set of particles (in red) are seen to lie roughly spread along the orbit (green line). As the underlying true state lies next to the edge of the FOV (black line), particles that are distributed close to it get higher weights. These particles (blue) are retained after resampling. The particle measurement update may expose the filter to the risk of weight depletion and covariance collapse, particularly when the sample size is small as a sizable fraction of particles may be presented with negligible likelihoods. Evaluating the measurement likelihoods can be avoided if the measurement update

is performed using a Kalman update step.

Hybrid Filter 2

Let t_e be the time at which the space object exits the observation range and $t_r > t_e$ be its time of re-entry. Then the ensemble of particles obtained at time t_r prior to the measurement update is essentially a sample drawn from the propagated state distribution $p(X_{t_r}|Z_{t_e})$. Let this ensemble be denoted by $A_{t_r} = \{X_{t_r}^1, X_{t_r}^2, \ldots, X_{t_r}^N\}$ where N represents the total number of particles. Then the mean and covariance of the PDF $p(X_{t_r}|Z_{t_e})$ may be approximated as

$$\begin{aligned} E(X_{t_r}|Z_{t_e}) &\approx \hat{\mu}_{A,t_r}, \\ cov(X_{t_r}|Z_{t_e}) &\approx \hat{C}_{A,t_r}. \end{aligned} \tag{3.14}$$

Here $\hat{\mu}_{A,t_r}$ and $\hat{C}_{A,t_r}$ are respectively the sample mean and sample covariance of A_{t_r}. It is then possible to obtain an approximate mean and covariance of the posterior PDF $p(X_{t_r}|Z_{t_r})$ by performing a Kalman measurement update on $\hat{\mu}_{A,t_r}$ and $\hat{C}_{A,t_r}$. The posterior state estimate obtained in this manner may then be appropriated for the subsequent UKF based estimation. The detailed steps involved in computing the

posterior mean μ_{X,t_r} and covariance C_{X,t_r} are given below:

$$
\begin{aligned}
\hat{\mu}_{A,t_r} &= \frac{1}{N}\sum_{l=1}^{N} X_{t_r}^{l} \\
\hat{C}_{A,t_r} &= \frac{1}{N-1}\sum_{l=1}^{N}\left(X_{t_r}^{l}-\hat{\mu}_{A,t_r}\right)\left(X_{t_r}^{l}-\hat{\mu}_{A,t_r}\right)^{T} \\
z_{t_r}^{l} &= g(X_{t_r}^{l}) \quad l=1,\ldots,N \\
\hat{z}_{A,t_r} &= \frac{1}{N}\sum_{l=1}^{N} z_{t_r}^{l} \\
\hat{C}_{ZZ} &= \frac{1}{N-1}\sum_{l=1}^{N}\left(z_{t_r}^{l}-\hat{z}_{A,t_r}\right)\left(z_{t_r}^{l}-\hat{z}_{A,t_r}\right)^{T}+R \\
\hat{C}_{XZ} &= \frac{1}{N-1}\sum_{l=1}^{N}\left(X_{t_r}^{l}-\hat{\mu}_{A,t_r}\right)\left(z_{t_r}^{l}-\hat{z}_{A,t_r}\right)^{T} \\
K_k &= \hat{C}_{XZ}\hat{C}_{ZZ}^{-1} \\
\mu_{X,t_r} &= \hat{\mu}_{A,t_r}+K_k\left(z_{t_r}-\hat{z}_{A,t_r}\right) \\
C_{X,t_r} &= \hat{C}_{A,t_r}-K_k\hat{C}_{ZZ}{K_k}^{T}
\end{aligned}
\tag{3.15}
$$

The Kalman measurement update enables the EnKF to keep the number of particles small even in high dimensional estimation problems. The estimation procedure followed by the hybrid filter 2 while outside the FOV and during the PF to UKF transition is similar to that of an EnKF [30, 31] i.e., using an ensemble of particles to perform uncertainty propagation and a Kalman measurement update for data assimilation. However, unlike hybrid filter 2, the EnKF performs the Kalman update on individual particles to obtain samples from the posterior PDF directly. Given an ensemble of states $\psi = [X_1, X_2, \cdots, X_n]$, corresponding measurement vectors $\phi = [h(X_1), h(X_2), \cdots, h(X_n)]$ and a recorded measurement y, the ensemble Kalman

update is given by

$$\hat{\psi} = \psi + K_k(Y - \phi), \tag{3.16}$$

where Y is the matrix of perturbed observations.

$$\begin{aligned} &Y = [y^1, \cdots, y^n], \;\; y^i = y + \nu^i \\ &\nu^i \sim \mathcal{N}(0, R). \end{aligned} \tag{3.17}$$

The posterior ensemble $\hat{\psi}$ is then used for propagation in the next time step. In contrast, the transition equations described in hybrid filter 2 computes the posterior mean and covariance directly from the statistics of the propagated state variable. An important advantage of this hybrid update is that it allows the number of particles to be varied during the filtering process. This is not possible in a standard EnKF in which the size of the ensemble is fixed at the beginning of the simulation.

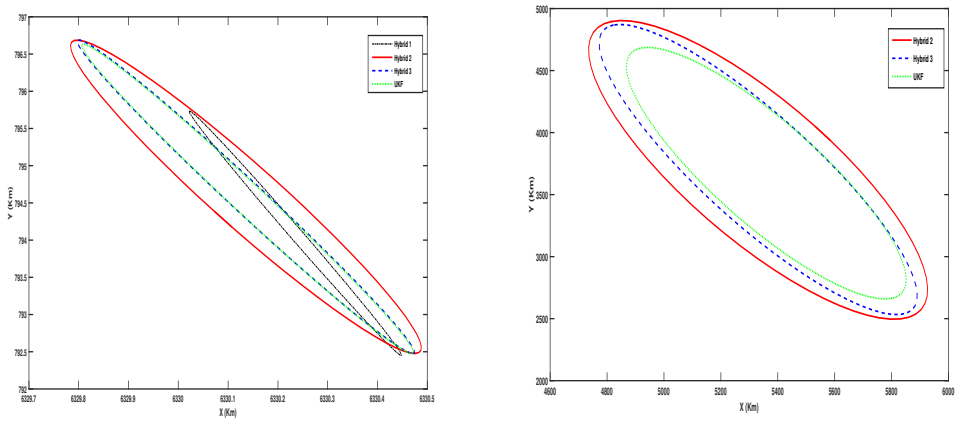
Hybrid Filter 3

The PF to UKF transition in hybrid filter 3 is also accomplished through a Kalman measurement update. However, unlike hybrid filter 2, it uses the UT to compute the Kalman gain and perform the mean and covariance update. To start with, it uses the ensemble averaged mean $\hat{\mu}_{A,t_r}$ and covariance $\hat{C}_{A,t_r}$ of the state vector to compute sigma points. Then the sigma points are used to compute the terms $\hat{z}$, $\hat{C}_{ZZ}$ and $\hat{C}_{XZ}$ with UT. The Kalman gain is computed using these matrices as $K_k = \hat{C}_{XZ}\hat{C}_{ZZ}^{-1}$. The transition is completed by evaluating the expressions for μ_{X,t_r} and C_{X,t_r} in equations 3.15 with these terms.

3.3.1 Effect of Transition Design

The following simulation was conducted to study the effect of different transition schemes on the posterior uncertainty estimate. A set of 1000 particles is sampled from an initial Gaussian PDF. The initial uncertainty in position is set to 1 km and that in velocity is set to 10 m/s along each direction. The particles are then propagated forward through 100 time steps with $\Delta t = 1s$. At this point, the first measurement z is registered and the moments of the posterior random variable are computed using the three different transition schemes. The 3-sigma ellipses of the posterior marginal distribution in X-Y coordinates computed using the three PF-UKF transition designs are presented in Figure 3.4a. The posterior covariance computed by propagating the initial Gaussian PDF with a UKF is also included for comparison. The covariance computed using hybrid filter 1 is seen to be the smallest among the four. This happens when the measurement likelihood drops considerably over short distances as a result of which a large section of the particles acquire negligible weights. The contribution of these particles to the posterior covariance estimate also reduces on account of their diminished weights. While it is true that one can employ a relatively small number of particles for estimation outside the FOV, as there are no measurements, relying on the particle measurement update during PF to UKF transition seems to once again bring about the curse of dimensionality. This indicates that the feasibility of a UKF-PF hybrid filter that uses a small number of particles without the risk of depletion as conceived in section 3.3 depends on the PF to UKF transition design. Transition design 1 employs the particle measurement update and is prone to underestimating the posterior covariance unless a sufficiently large particle ensemble is used.

The risk of covariance collapse is averted when the transition is performed



(a) (b)

Figure 3.4: Effect of transition design on the estimated uncertainty (C_{X_t}) in posterior PDF. Reprinted with permission from [6].

through a Kalman update on the ensemble. As Figure 3.4a indicates, the posterior covariance estimates given by transition designs 2 and 3 are much larger in comparison to that given by design 1. Transition designs 2 and 3 are not subject to weight depletion as they do not rely on measurement likelihoods to perform the update. In this case, the covariance estimated by the UKF is seen to be similar in size to that estimated by hybrid filters 2 and 3. However, on raising the initial uncertainty in velocity to 1km/s and increasing the simulation time to 500 time steps, the uncertainty estimated using the UKF is seen to be distinctly smaller in comparison to hybrid filters 2 and 3. This is shown in Figure 3.4b. The hybrid filter 1 is observed to undergo complete covariance collapse in this case. Hence, it is not included in Figure 3.4b. It clearly appears that, by transitioning from PF to UKF through a Kalman update, the problem of particle depletion and subsequent covariance collpase can be avoided. A general algorithm for implementing the UKF-PF hybrid filter is presented in algorithm 1.

Algorithm 1 UKF-PF Hybrid filter for space object tracking

S_1 : PDF in functional form (inside FOV), S_2 : PDF as ensemble (outside FOV),

$C(X)$: Boundary of FOV, P_d : Probability of detection.

Initialize: $P(X) = P_0(X), S = S_0$

At t_k

if $S = S_1$ **then**
  if $C(X_k) <= 0$ **then**
    Use UKF
    SET $S = S_1$
  else
    SAMPLE FROM P(X)
    USE PF
    SET $S = S_2$
  end if
else
  if $C(X_k) <= 0$ **and** $\eta > \eta_0$ **then**
    EXECUTE PF-UKF TRANSITION
    COMPUTE P(X)
    SET $S = S_1$
  end if
else
  USE PF
  SET $S = S_2$
end if

3.4 Numerical Simulations

In this section, the UKF-PF hybrid filtering framework is applied to two test case problems of space objects in low earth orbits (LEO). The three hybrid filter variants are simulated along with standard implementations of UKF and PF to compare the estimation performance. The estimation results are assessed for accuracy and consistency.

The accuracy of the estimator is evaluated in terms of the root mean squared error (RMSE) in position estimates. The RMSE in position estimate at time t may be calculated as

$$RMSE_{position} = \sqrt{\frac{1}{N_{Mo}} \sum_{i=1}^{N_{Mo}} \sum_{j=1}^{3} (x'_{j,t,i} - X_{j,t,i})^2}. \tag{3.18}$$

Here N_{Mo} represents the number of Monte Carlo runs over which the RMSE is computed. The terms $x'_{j,t,i}$ and $X_{j,t,i}$ represent the actual and estimated position coordinates of the object at time t in the direction j during the i^{th} Monte Carlo run. Smaller RMSE values represent more accurate estimates.

The normalized estimation error squared (NEES) test is employed to evaluate the consistency of estimates [32]. The NEES test is performed by computing the normalized residual $\beta_{t,i}$, which is defined as

$$\beta_{t,i} = ({x'}_{t,i} - X_{t,i})^T {\mathbf{C}_{\mathbf{X,t,i}}}^{-1} ({x'}_{t,i} - X_{t,i}). \tag{3.19}$$

Here ${x'}_{t,i}$ represents the actual state occupied by the object at time t in the i^{th} Monte Carlo run. If $X_{t,i} \in \mathbb{R}^n$ is distributed according to a Gaussian PDF, then $\beta_{t,i}$ is a chi-square random variable with expected value n. To assess the consistency of the hybrid filter, the average NEES test statistic computed over multiple Monte

Carlo runs is considered. The average NEES test statistic for the estimates at time t computed over N_{Mo} simulations is given by

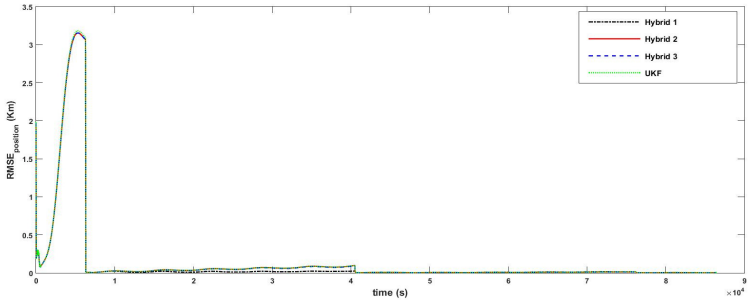
$$\beta_t = \frac{1}{N_{Mo}} \sum_{i=1}^{N_{Mo}} \beta_{t,i}. \tag{3.20}$$

When the state vector is a six dimensional Gaussian random variable, the sum $N_{Mo}\beta_t$ can be shown to be distributed according to a χ^2 density with $6N_{Mo}$ degrees of freedom. Consequently, the consistency of the estimator may be tested by examining whether β_t falls within probable bounds computed from the corresponding χ^2 distribution. When the average NEES is computed over 50 Monte Carlo runs, a 99.5 percent probability upper bound for the random variable β_t is found to $Ub_{0.995} = 7.3369$. If β_t assumes a value above 7.3369, then the covariance estimates are inconsistent with the estimation errors, i.e., it is likely that the covariances $C_{X,t,i}$ are underestimated. In other words, the estimates are optimistic.

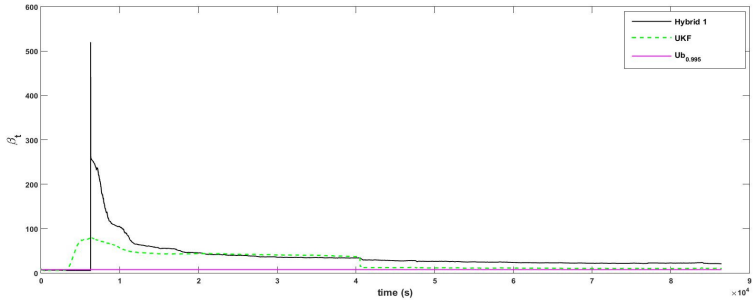
Case 1: In this test case, the estimated initial state of the object is set at

$$X_0 = \left[7800 \;\; 0 \;\; 0 \;\; 0 \;\; 6.8443\cos(\pi/4) \;\; 6.8443\sin(\pi/4)\right]^T, \tag{3.21}$$

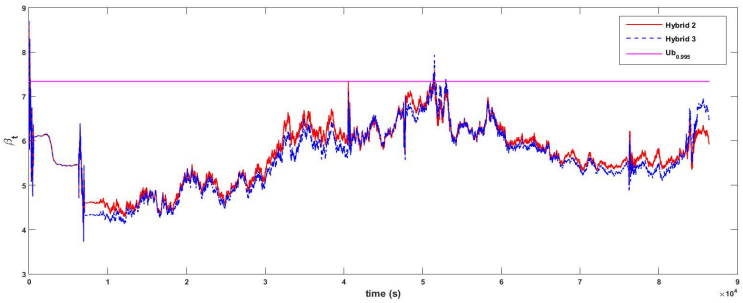
where the lengths and speeds are in km and km/s respectively. This is a 45° inclined LEO with a period of 6080 s and eccentricity of 0.0833. The uncertainty in the initial state estimate is characterized by a standard deviation of 1 km in position estimates and 1 m/s in velocity estimates. The probability of detection is set at 0.9. The filters are employed to estimate the state of the space object for a total duration of 24 hours. In hybrid filter simulations, the number of particles used during the sequential Monte Carlo update is set at 500. A sequential importance resampling(SIR) filter is used as the standard PF implementation [24]. In order to maintain uniform computational costs, the SIR is also implemented with 500 particles.



(a) $RMSE_{position}$



(b) NEES Plot 1



(c) NEES Plot 2

Figure 3.5: Monte Carlo averaged tracking results for LEO object test case 1. Reprinted with permission from [6].

The simulations for test case 1 are repeated over 50 Monte Carlo runs and the averaged values for NEES and RMSE are computed. The estimation results for the

hybrid filters and the UKF for test case 1, are plotted in Figure 3.5. In Figure 3.5a the RMSE in position for each filter are plotted against time. As the objects are initialized inside the FOV, the uncertainty in position estimates are observed to diminish sharply in the beginning.

Once the object moves out of the FOV, measurements become unavailable, errors accumulate and the amount of uncertainty increases steadily as signified by the upswing in the $RMSE_{position}$ plots. However, after reentry into the FOV, more information is added with each recorded measurement and the magnitude of RMSE drops again. This pattern repeats over the many subsequent FOV entries and exits. All four filters follow this trend and showcase similar performance, as seen in Figure 3.5a. For the three hybrid filters and the UKF, the error in position estimates during the last four hours of the simulated time are seen to be of the order of 10^0m.

Figure 3.5b shows the results of NEES test for hybrid filter 1 and the UKF for test case 1. A horizontal line indicating $y = Ub_{0.995}$ has been included for reference. It is seen that the estimates generated by hybrid filter 1 and UKF are inconsistent for a very long time. Comparing the NEES plots of hybrid filter 1 and UKF with their corresponding RMSE plots reveals the following. In the case of UKF, once the object exits the FOV, the estimation error and the value of NEES test statistic are both seen to grow. The NEES test statistic is seen to overstep the $Ub_{0.995}$ line during this stage. This indicates that when the object is outside the FOV, the covariance estimated by the UKF does not grow fast enough to ensure that the estimates remain consistent. However, the value of UKF NEES test statistic is seen to drop once the measurements start to reappear. In contrast, the hybrid filter 1 estimates are seen to remain consistent throughout the flight of the Object outside the FOV until the first re-entry. The NEES plot of hybrid filter 1 crosses the $Ub_{0.995}$ only when the measurements reappear and the PF to UKF transition is triggered. As mentioned

before, hybrid filter 1 relies on the particle measurement update which is prone to underestimating the posterior uncertainty. As a result, the NEES test statistic is seen to spike during the transition. The NEES test results for the hybrid filters 2 and 3 are plotted in Figure 3.5c. The results show that estimates provided by hybrid filters 2 and 3 remain consistent during most of the simulated time.

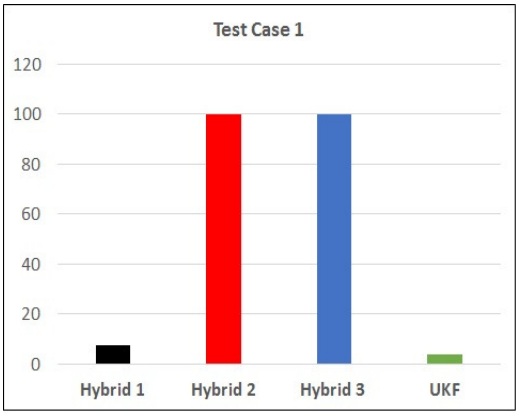


Figure 3.6: Consistency of filtered estimates in test case 1. Reprinted with permission from [6].

The performance of the filters in NEES test may be compared using the fraction of the total simulated time period during which each filter generated consistent estimates. The fraction of times during which the hybrid filters and the UKF provided estimates that lie within the 99.5 percent bounds in test case 1 are plotted in Figure 3.6. It is seen that estimates provided by hybrid filter 2 and 3 are consistent during 99.84 percent and 99.69 percent of the simulated time in test case 1. Estimates provided by the hybrid filter 1 are seen to be consistent during 7.28 percent of the simulated time. The consistency fraction for the UKF in test case 1 is found to be 3.69 percent. This indicates that UKF and hybrid filter 1 are prone to underestimating the posterior covariance, even though the estimation errors committed by

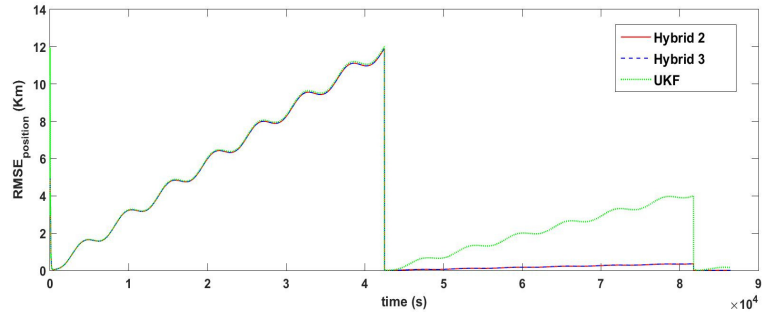
all four filters are similar.

When it was employed with 500 particles to estimate the test case 1, the covariance of the SIR filtered is observed to collapse within a few time steps. The small process noise and accurate measurements result in loss of diversity of the ensemble at each resampling step. Consequently, the sample covariance becomes negligible in a few time steps when the actual estimation errors are significant. As a result, the value of β_t for the SIR filter is found to quickly blow up. It may well be possible to perform estimation of a six dimensional nonlinear system using an SIR filter when implemented with a much larger number of particles. However increasing the number of particles to prevent weight depletion will also increase the computational cost. It is notable that the hybrid filters 2 and 3 offer reliable performance in this six dimensional estimation problem while requiring only a relatively small number of particles.

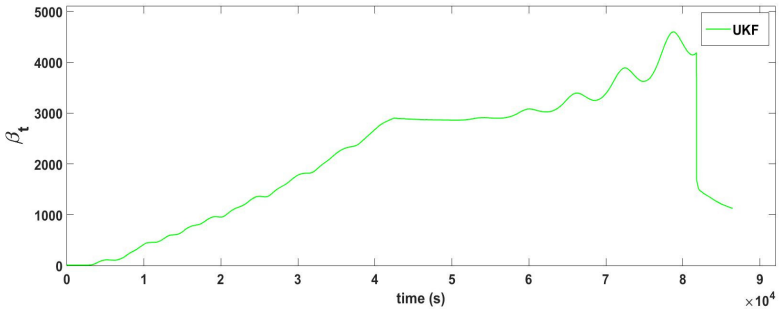
Case 2: In this case, the estimated initial state of the object is set to

$$X_0 = \left[6800 \;\; 0 \;\; 0 \;\; 0 \;\; 7.5989\cos(\pi/3) \;\; 7.5989\sin(\pi/3)\right]^T. \tag{3.22}$$

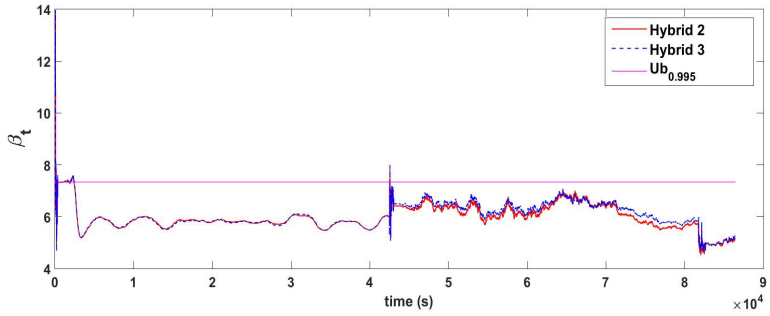
This is a 60^o inclined low earth orbit with a time period of 5580.5 s. The initial uncertainty in position is increased to 3 km along each direction and that in velocity is raised to 1 km/s. The probability of detection is reduced to 0.4. As mentioned before, for space objects, the growth in uncertainty is remarkably sensitive to the initial uncertainty in velocity. When the uncertainty is larger and state PDF is more diffuse, the accumulation of errors due to linearization also becomes much more severe. The nonlinearity induced distortion of state PDF is also larger in such a case. Case 2 is used to test the performance of the hybrid filters under such conditions. The simulation was conducted for a period of 24 hrs. The results, averaged over 50



(a) $RMSE_{position}$



(b) NEES Plot 1



(c) NEES Plot 2

Figure 3.7: Monte Carlo averaged tracking results for LEO object test case 2. Reprinted with permission from [6].

Monte Carlo runs, are given in Figure 3.7. The estimation results for hybrid filter 1 in test case 2 are not presented as it was observed to undergo covariance collapse due to significant particle depletion. The SIR filter is also observed to undergo covariance collapse in test case 2.

The $RMSE_{position}$ plots given in Figure 3.7a indicate that for all three filters, the errors in position estimates quickly drop in the beginning since measurements are recorded. Once the object exits the FOV, the errors start to grow. The errors shrink again when the object reenters the FOV of the sensor and a new batch of measurements become available.

From Figure 3.7a it can be observed that the estimation errors committed by UKF in test case 2 are seen to be much worse than that by hybrid filters 2 and 3 . For hybrid filters 2 and 3, the errors in position estimates during the final 4 hours of the simulated time are seen to be of the order of 10^{-1} Km. For UKF, this number is seen to be of the order of 10^{0} Km. The NEES test results for the UKF estimates in test case 2 are plotted in Figure 3.7b. The results indicate that the UKF estimates are inconsistent during most of the simulated time. By comparing Figure 3.7b with Figure 3.7a, it can be observed that the value of UKF NEES test static grows as the estimation error increases. Once the $Ub_{0.995}$ is overstepped, the UKF estimates stay inconsistent for the entire length of the remaining simulation time, in spite of recording several additional measurements. The NEES test results for hybrid filters 2 and 3 are plotted in Figure 3.7c. The hybrid filters 2 and 3 are seen to produce consistent estimates during 98.41 and 98.49 percent of the times respectively. In contrast, the UKF estimates are consistent during only 2.74 percent of the total simulation time. The fraction of the simulated time during which each filter offered consistent estimates is plotted in Figure 3.8.

Our results in this chapter indicate how the unimodal Gaussian filters such as the EKF and UKF can prove to be inadequate in representing the state PDF in the presence significant nonlinearity. We also found that while the particle filter does not enforce restrictive assumptions on the nature of PDF, its implementation becomes computationally expensive as the dimensionality of the state space increases.

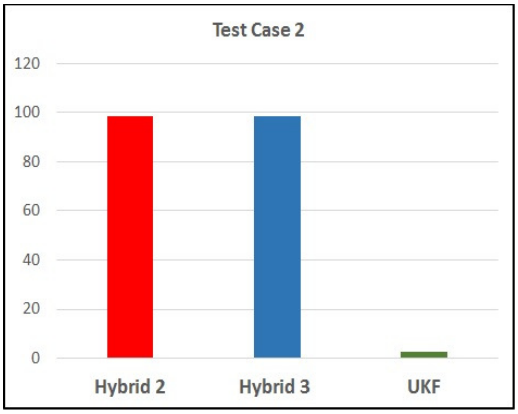


Figure 3.8: Consistency of filtered estimates in test case 2. Reprinted with permission from [6].

Based on these observations, a general multimodal nonlinear filter named Particle Gaussian Mixture-I (PGM-I) Filter has been proposed by the authors. The PGM-I filter employs a particle approximation for propagation and a Kalman update of the type employed in hybrid filter 2 or 3 for measurement update. Additionally, it does not require the assumption 2 enforced on process noise. The PGM filtering algorithm is developed in detail in the next chapter.

4. PARTICLE GAUSSIAN MIXTURE FILTERS -I*

In this chapter, we propose a particle Gaussian mixture-I (PGM-I) filter [2, 1] for nonlinear estimation. The PGM-I filter design is an extension of the UKF-PF hybrid filter that has been proposed for space object tracking [5] in the previous chapter. The PGM-I filter is conceived to keep track of the nonlinear uncertainty propagation, without performing any additional optimization or splitting operations during the propagation step. For ease of treatment and clarity of exposition, we shall not consider the measurement update aspect of the GMM filtering problem in this chapter, which will be treated in chapter 5. We make the following assumption for the remainder of this chapter.

Assumption 3. *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 Eq. 1.13 after an observation z_n is approximated arbitrarily well by the Least Squares/ Kalman update equations. 1.17-1.18.*

4.1 The Particle Gaussian Mixture (PGM) Filter

In this section, we first present the PGM filter. The basic assumption underlying the PGM algorithm is that the predicted prior and posterior filter densities can be

*Parts of this chapter were reprinted with permission from [1, 2].

represented using a GMM. In particular, let:

$$\pi_n^-(x) = \sum_{i=1}^{M^-(n)} \omega_i^-(n)\mathcal{G}_i^-(x;\mu_i^-(n), P_i^-(n)), \tag{4.1}$$

$$\pi_n(x) = \sum_{i=1}^{M(n)} \omega_i(n)\mathcal{G}_i(x;\mu_i(n), P_i(n)). \tag{4.2}$$

In general, $M^-(n)$ and $M(n)$ need not be the same, however, owing to Assumption 3, they are assumed to be equal for the purposes of this chapter. 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

---

Algorithm 2 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 $\pi_{n-1}(.)$.
   (b) Sample $X^{(i)}$ from $p(.|X^{(i)'})$.
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 $\{w_i^-(n), \mu_i^-(n), P_i^-(n)\}$.
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 equations 1.17, 1.18.
4. n = n+1, go to Step 1.

---

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 $\{x_n^1, \cdots, x_n^i, \cdots, x_n^{N_p}\}$ 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 $Sw_n = \{w_n^1, \cdots, w_n^i, \cdots, w_n^{N_p}\}$. Let

$$x_{n+1}^{i-} = f(x_n^i) + w_n^i. \tag{4.3}$$

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 [33, 34]. A brief description of clustering is provided in Appendix A. In the experimental results presented in this chapter, we use the simple k-means clustering algorithm [35], 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 $G_M^* = \{S_1, \cdots, S_M\}$ such that

$$G_L^* = \underset{G_L}{\operatorname{argmin}} \sum_{i=1}^{M} \sum_{x_j \in S_i} \|x_i - \mu_i\|^2. \tag{4.4}$$

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.

$$n_i = \sum_{j=1}^{N} \mathbb{1}(x_j \in S_i), \quad w_i = \frac{n_i}{N}, \tag{4.5}$$
$$\mu_i = \frac{1}{n_i} \sum_{x_j \in S_i} x_j, \quad C_i = \frac{\sum\limits_{x_j \in S_i} (x_j - \mu_i)(x_j - \mu_i)^T}{n_i - 1}.$$

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

3) Measurement update: Incorporate the measurement information by updating the means and covariances of all M modes individually using a least squares/ Kalman measurement update. Also update the mixture weights using the mode likelihoods $l_i(n)$ as in equation 1.12. In the present work we have considered two different approaches to computing the covariance terms $\big(P_{i,ZX}(n+1), P_{i,ZZ}(n+1)\big)$ and the expectations $\Big(E_i\big(h(X)\big)\Big)$ 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 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 $S_{j,n+1}^{-1} = \{x_{j,n+1}^{1-}, \cdots, x_{j,n+1}^{i-}, \cdots, x_{j,n+1}^{Nj-}\}$ 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 $Y_{j,n+1}^{-1} = \{h(x_{j,n+1}^{1-}), \cdots, h(x_{j,n+1}^{i-}), \cdots, h(x_{j,n+1}^{Nj-})\}$.

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

4.1.1 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 4. *We assume that there exists $C < \infty$ and $\rho < 1$ such that:*

$$||F_{z_n}(F_{z_{n-1}}(..(F_{z_1}(\pi_0))..)) - F_{z_n}(F_{z_{n-1}}(..(F_{z_1}(\pi'_0))..))|| \leq C\rho^n||\pi_0 - \pi'_0||, \tag{4.6}$$

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

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

Assumption 5. *Let $Prob(||\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})|| > \epsilon) < \delta$, for all n. Further, we assume that $Prob(||\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})|| > M) = 0$, 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).*

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 4, it follows that* $||\hat{\pi}_n - \pi_n|| \leq \frac{(C+1)\epsilon}{1-\rho}$.

Proof. We have:

$$\begin{aligned}
\hat{\pi}_n - \pi_n &= \hat{F}_{z_n}(\hat{F}_{z_{n-1}}(..(\hat{F}_{z_1}(\pi_0))..)) \\
&\qquad - F_{z_n}(F_{z_{n-1}}(...(F_{z_1}(\pi_0))..)), \\
&= \underbrace{[\hat{F}_n(\hat{F}_{n-1}(..(\hat{F}_1(\pi_0))..)) - F_n(\hat{F}_{n-1}(..(\hat{F}_1(\pi_0))..))]}_{\Delta_n} \\
&+ \underbrace{[F_n(\hat{F}_{n-1}(..(\hat{F}_1(\pi_0))..)) - F_n(F_{n-1}(\hat{F}_{n-2}(..(\hat{F}_1(\pi_0))..)))]}_{\Delta_{n-1}} \\
&+ \cdots + \underbrace{[F_n(F_{n-1}(....(\hat{F}_1(\pi_0))...)) - F_n(..(F_1(\pi_0))..)]}_{\Delta_1}. \qquad (4.7)
\end{aligned}$$

Note that the different terms on the RHS above are:

$$\begin{aligned}
\Delta_n &= \hat{F}_n(\hat{\pi}_{n-1}) - F_n(\hat{\pi}_{n-1}), \\
\Delta_{n-1} &= F_n(\hat{F}_{n-1}(\hat{\pi}_{n-2})) - F_n(F_{n-1}(\hat{\pi}_{n-2})), ... \\
\Delta_1 &= F_n(..(F_2(\hat{F}_1(\pi_0)))..) - F_n(..(F_2(F_1(\pi_0)))..). \qquad (4.8)
\end{aligned}$$

Using Assumption 4 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,

$$||\hat{\pi}_n - \pi_n|| \leq \sum_{i=1}^{n-1} C\rho^{n-i}\epsilon + \epsilon \leq \frac{(C+1)\epsilon}{1-\rho}. \qquad (4.9)$$

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. $\square$

Lemma 2. *Let Assumptions 4 and 5 hold. Given any $\delta, \nu > 0$, there exists an $\bar{N} < \infty$, such that:*

$$Prob(||\hat{\pi}_n - \pi_n|| > \frac{(1+\nu)(1+C)\epsilon}{1-\rho}) \leq \bar{N}\delta, \tag{4.10}$$

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

Proof. Let $e_n = ||\hat{\pi}_n - \pi_n||$, and let $\epsilon_k = ||\hat{F}_{z_k}(\hat{\pi}_{k-1}) - F_{z_k}(\hat{\pi_{k-1}})||$. It follows that $e_n \leq \sum_{k=-\infty}^{n} C\rho^{n-k}\epsilon_k$. Choose n' such that $\sum_{i=-\infty}^{n'} C\rho^{n-i} \leq f$, where $f = \frac{\nu(C+1)\epsilon}{M(1-\rho)}$. Then:

$$e_n = \underbrace{\sum_{k=n'}^{n} C\rho^{n-k}\epsilon_k}_{\bar{e}_n} + \underbrace{\sum_{k=-\infty}^{n'} C\rho^{n-k}\epsilon_k}_{\Delta_n^*}. \tag{4.11}$$

From Assumption 5, it follows that $Prob(||\Delta_n^*|| > fM) = 0$, and thus,

$$Prob(||\Delta_n|| > \frac{\nu(C+1)\epsilon}{1-\rho}) = 0. \tag{4.12}$$

Similarly, from Lemma 1, it follows that:

$$Prob(\bar{e}_n > \frac{(C+1)\epsilon}{1-\rho}) \leq (n-n')\delta \equiv \bar{N}\delta. \tag{4.13}$$

Using the equations 4.12 and 4.13, it follows that $Prob(e_n > \frac{(1+\nu)(C+1)\epsilon}{1-\rho}) \leq \bar{N}\delta$. $\square$

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,we define the following:

$$P(\hat{\pi}_{n-1}) \equiv \hat{\pi}_n^- = \sum_{i=1}^{M^-(n)} \hat{\omega}_i^-(n)\mathcal{G}_i(x; \hat{\mu}_i^-(n), \hat{P}_i^-(n)), \tag{4.14}$$

$$\hat{P}(\hat{\pi}_{n-1}) \equiv \hat{\hat{\pi}}_n^- = \sum_{i=1}^{M^-(n)} \hat{\hat{\omega}}_i^-(n)\mathcal{G}(x; \hat{\hat{\mu}}_i^-(n), \hat{\hat{P}}_i^-(n)), \tag{4.15}$$

$$F_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\omega}_i(n)\mathcal{G}(x; \hat{\mu}_i(n), \hat{P}_i(n)), \tag{4.16}$$

$$\hat{F}_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\hat{\omega}}_i(n)\mathcal{G}(x; \hat{\hat{\mu}}_i(n), \hat{\hat{P}}_i(n)). \tag{4.17}$$

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 $\epsilon' > 0$, andy $\delta' > 0$, 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:*

$$Prob(|\hat{\hat{\omega}}_i^-(n) - \hat{\omega}_i^-(n)| > \epsilon') < \delta', \tag{4.18}$$

$$Prob(||\hat{\hat{\mu}}_i^-(n) - \hat{\mu}_i^-(n)||_2 > \epsilon') < \delta', \tag{4.19}$$

$$Prob(||\hat{\hat{P}}_i^-(n) - \hat{P}_i^-(n)|| > \epsilon') < \delta', \tag{4.20}$$

for all i.

Proof. Given a random variable (r.v) X, and a function $f(X)$ such that $E(f(X)) = \bar{f}$, and $Var(f(X)) = \sigma_f^2$, let $\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$, where X_i are samples of the r.v. X. For large enough N, it follows from the Central Limit Theorem [36] that:

$$P(|\hat{f} - \bar{f}| > \epsilon) < 1 - \int_{-\epsilon}^{\epsilon} \frac{\sqrt{N}}{\sqrt{2}\pi\sigma_f} e^{-\frac{x^2 N}{2\sigma_f^2}} dx. \tag{4.21}$$

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 equation 4.18 is satisfied.

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

$$P\{\Big| \|X\|_2 - \sqrt{d} \Big| \geq t\} \leq 2e^{\frac{-ct^2}{k^4}}, \forall t \geq 0. \tag{4.22}$$

Here c is a positive constant and $k = \max_i \|x_i\|_{\psi_2}$ where $\|x_i\|_{\psi_2} = \inf\{t; E\{e^{(\frac{x_i}{t})^2}\} \leq 2\}$ [37]. If $\hat{\hat{\mu}}_i^-(n)$ is determined as the sample mean of points assigned to cluster C_i, then from equation 4.22 we have

$$P\{\Big| \Big\| (\hat{P}_i^-(n)^{\frac{-1}{2}} \sqrt{N_j})(\hat{\hat{\mu}}_i^-(n) - \hat{\mu}_i^-(n)) \Big\|_2 - \sqrt{d} \Big| \geq t\} \leq 2e^{\frac{-ct^2}{k^4}} \tag{4.23}$$

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

$$P\left\{ \Big\| \left(\hat{\hat{\mu}}_i^-(n) - \hat{\mu}_i^-(n)\right) \Big\|_2 \geq \frac{t + \sqrt{d}}{\Big\| \left(\hat{P}_i^-(n)\right)^{\frac{-1}{2}} \Big\|_2 \sqrt{N_j}} \right\} \leq 2e^{\frac{-ct^2}{k^4}}, \tag{4.24}$$

Using equation 4.24 and by setting

$$\frac{t+\sqrt{d}}{\left\|(\hat{P}_i^-(n))^{\frac{-1}{2}}\right\|_2\sqrt{N_j}} < \epsilon', \tag{4.25}$$

$$2e^{\frac{-ct^2}{k^4}} < \delta', \tag{4.26}$$

it is possible to choose an N_j so that equation 4.19 is satisfied. The minimum value of N_j that satisfies the above set of equations is chosen as $N_{\epsilon',\delta'}(\mu_j)$.

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

$$P\{\frac{\left\|\hat{\hat{P}}_i^-(n)-\hat{P}_i^-(n)\right\|}{\left\|\hat{P}_i^-(n)\right\|} \geq ck^2(\sqrt{\frac{d+u}{N_j}}+\frac{d+u}{N_j})\} \leq 2e^{-u}, \tag{4.27}$$

where k is a constant greater than or equal to one and $c \geq 0$[37]. Here $\|.\|$ represents the spectral norm of the covariance matrix. Hence, by picking an N_j such that

$$ck^2(\sqrt{\frac{d+u}{N_j}}+\frac{d+u}{N_j})\left\|(\hat{P}_i^-(n)\right\| < \epsilon', \tag{4.28}$$

$$2e^{-u} < \delta', \tag{4.29}$$

the condition given in equation 4.20 can be satisfied. Let$N_{\epsilon',\delta'}(C_j)$ be the minimum N_j that satisfies this condition. Pick $N_{\epsilon',\delta'}(j) = \max\left(N_{\epsilon',\delta'}(\omega_j), \frac{N_{\epsilon',\delta'}(\mu_j)}{\hat{\omega}_j^-}, \frac{N_{\epsilon',\delta'}(P_j)}{\hat{\omega}_j^-}\right)$. Such $N_{\epsilon',\delta'}(j)$ can be found for all clusters C_j and given that we choose $N_{\epsilon',\delta'}$ as:

$$N_{\epsilon',\delta'} = \max_j N_{\epsilon',\delta'}(j),, \tag{4.30}$$

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. $\square$

It may be shown that under Assumption 3 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 Eq. 4.30. 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:*

$$Prob(|\hat{\hat{\omega}}_i(n) - \hat{\omega}_i(n)| > K(n)\epsilon') < \delta', \tag{4.31}$$

$$Prob(\left\|\hat{\hat{\mu}}_i(n) - \hat{\mu}_i(n)\right\| > K(n)\epsilon') < \delta', \tag{4.32}$$

$$Prob(\left\|\hat{\hat{P}}_i(n) - \hat{P}_i(n)\right\| > K(n)\epsilon') < \delta', \tag{4.33}$$

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

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.

$$\begin{aligned}\hat{\hat{\pi}}(x) - \hat{\pi}(x) &= \frac{1}{(2\pi)^{d/2}|\hat{\hat{P}}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{\hat{P}}^{-1}(x-\mu)} - \\ &\quad \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)}, \\ &\approx \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \hat{P}^{-1}(x-\mu)} \\ &\quad \times \frac{1}{2}(x-\mu)^T(\hat{P}^{-1}\Delta\hat{P}^{-1})(x-\mu), \end{aligned} \tag{4.34}$$

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

$$e^{-\frac{1}{2}(x-\mu)^T(\hat{P}+\Delta)^{-1}(x-\mu)} \approx e^{-\frac{1}{2}(x-\mu)^T\hat{P}^{-1}(x-\mu)} e^{-\frac{1}{2}(x-\mu)^T\hat{P}^{-1}\Delta\hat{P}^{-1}(x-\mu)}, \tag{4.35}$$

which in turn implies equation 4.34. This in turn implies that:

$$\begin{aligned}||\hat{\hat{\pi}} - \hat{\pi}|| &\approx \frac{1}{(2\pi)^{d/2}|\hat{P}|^{1/2}} \times \int e^{-\frac{1}{2}(x-\mu)^T\hat{P}^{-1}(x-\mu)} \frac{1}{2}(x-\mu)^T\hat{P}^{-1}\Delta\hat{P}^{-1}(x-\mu)dx, \\ &\leq \frac{C(\hat{P})\epsilon'}{(2\pi)^{d/2}|\hat{P}|^{1/2}} \times \int e^{-\frac{1}{2}(x-\mu)^T\hat{P}^{-1}(x-\mu)} \frac{1}{2}(x-\mu)^T\hat{P}^{-1}(x-\mu)dx, \end{aligned} \tag{4.36}$$

since there exists $C(\hat{P}) < \infty$ such that

$$\frac{1}{2}(x-\mu)^T\hat{P}^{-1}\Delta\hat{P}^{-1}(x-\mu) \leq C(\hat{P})\epsilon'\frac{1}{2}(x-\mu)^T\hat{P}^{-1}(x-\mu) \tag{4.37}$$

owing to the fact that $||\Delta|| < \epsilon'$.

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

$$||\hat{\hat{\pi}} - \hat{\pi}|| \leq C(\hat{P})\epsilon' \frac{1}{(2\pi)^{d/2}} \int e^{-1/2y^T y} y^T y dy = C(\hat{P})\epsilon' d. \tag{4.38}$$

The last step in the above equation follows from noting that $Y'Y$ 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 7 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:*

$$C(n)K(n)\epsilon'(n)d = \epsilon, \tag{4.39}$$

$$\delta'(n) = \frac{\delta}{N}, \tag{4.40}$$

and the corresponding number of samples $N_{\epsilon'(n),\delta'(n)}(n)$ be chosen according to equation 4.30, then it follows that $||Prob||\hat{F}_{z_n}(\hat{\pi}_{n-1}) - F_{z_n}(\hat{\pi}_{n-1})|| > \epsilon) \leq \frac{\delta}{N}$.

Proof. Recall that $\hat{\pi}_n = F_{z_n}(\hat{\pi}_{n-1})$, and $\hat{\hat{\pi}}_n = \hat{F}_{z_n}(\hat{\pi}_{n-1})$. Then, from Lemma 7 we have that $||\hat{\pi}_n - \hat{\hat{\pi}}_n|| \leq C(n)K(n)d\epsilon'(n)$ if $|\hat{\theta}_i(n) - \hat{\hat{\theta}}_i(n)| < \epsilon'(n)$ for all i, where $\hat{\theta}_i(n)$ represents the true parameters underlying the GMM representation of $\hat{\pi}_n$ and $\hat{\hat{\theta}}_i(n)$

represents their PGM approximation. Hence:

$$Prob(||\hat{\pi}_n - \hat{\hat{\pi}}_n|| > C(n)K(n)d\epsilon'(n)) < \delta'(n), \tag{4.41}$$

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 $Prob(||\hat{\hat{\pi}}_n - \hat{\pi}_n|| > \frac{(1+\nu)C\epsilon}{1-\rho}) \leq \delta$, for all n for any arbitrarily small $\epsilon, \delta, \nu > 0$. However, in order for Assumption 5 to be valid, the sample averages $\hat{\hat{\theta}}_n$ have to be almost surely bounded. Here $\hat{\hat{\theta}}_n$ 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 $\hat{\hat{\theta}}_n^N \to \hat{\theta}_n$ as $N \to \infty$, where $\hat{\hat{\theta}}_n^N$ represent the estimate of the parameters after N samples. Given the sample size is large enough, the estimate $\hat{\hat{\theta}}_n^N$ 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 3 and 4 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 equation 4.39, the required confidence $\delta'(n)$ from equation 4.40, and the corresponding minimum number of samples $N_{\epsilon'(n),\delta'(n)}$ from equation 4.30, then:*

$$Prob(||\hat{\hat{\pi}}_n - \hat{\pi}_n|| > \frac{(1+\nu)C\epsilon}{1-\rho}) \leq \delta. \tag{4.42}$$

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 3, 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 chapter 5.*

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. This is a fact that is typically ignored in other mixture filters such as the PF and the GMF.*

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, equations 4.25, 4.28 indicate that the number of samples required to estimate the component means and covariances increase only as $O(d)$. From the above analysis, equation 4.30, 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: equation 4.39 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, equation 4.21*

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.*

4.1.2 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 [26, 31] 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 of dimensionality. Consider the simple one dimensional example shown in Figure 4.1. 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 Figure 4.1. Please see the references [25, 38, 39] 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[30, 26] 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[31]. 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 [40] 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 [19]. 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

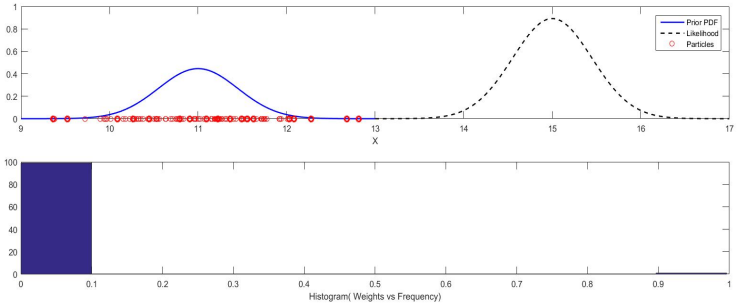


Figure 4.1: Particle Depletion

GMM using the Fokker-Planck equation [20] 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 [21]. The problem of covariance splitting during propagation can be avoided given that the mixand covariances are sufficiently small. The blob filter of [22] 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 not the 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 $\{\omega_i, \mu_i, C_i, i = 1 \cdots G\}$. 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 $i = 1 \cdots G$, the GSPF samples a set of M points $x_{ni}^j, \{j = 1 \cdots M\}$ 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 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.

4.2 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})[21] 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

$$L_{mes}(\theta_{a,M}) = \sum_{i=1}^{N_p} \pi_{\theta^a}(x^{i-}_{n+1}) \tag{4.43}$$

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_{mex} 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 [35]. Lloyd's algorithm finds a local minimum to the sum of squares of within-group distances appearing in equation 4.4. The time complexity of the Lloyd's algorithm is known to be $O(N_pMdi)$ 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 [41]. Implementing the naive clustering strategy as described here will result in a quadratic time complexity in $M^{-}_{max}(n+1)$.

Merging: Depending on the clustering scheme, dynamics and measurement models,

Algorithm 3 Clustering Strategy

Input: $S_{n+1}^{-1} = \{x_{n+1}^{1-}, \cdots, x_{n+1}^{i-}, \cdots, x_{n+1}^{N_p-}\}, M_{max}^{-}(n+1)$
Output: $\theta_{a^*,M*}$, $M* \leq M_{max}^{-}(n+1)$

$M \leftarrow M_{max}^{-}(n+1)$
$\theta_{a^*,M*} \leftarrow \theta_{a,M_{max}^{-}(n+1)}$
$L_{mes}^{*} \leftarrow L_{mes}(\theta_{a,M_{max}^{-}})$
while $M > 1$ **do**
  $M \leftarrow M - 1$
  Compute $\theta_{a,M}$ using k-means
  if $L_{mes}(\theta_{a,M}) \geq L_{mes}^{*}$ **then**
    $\theta_{a^*,M*} \leftarrow \theta_{a,M}$
    $L_{mes}^{*} \leftarrow L_{mes}(\theta_{a,M})$
  end if
end while

one may observe several closely distributed mixture modes in the posterior PDF. 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 [42] as a measure of similarity between modes i and j.

$$D(i,j) = \frac{\int (\mathcal{G}_i(x,\mu_i,P_i) - \mathcal{G}_j(x,\mu_j,P_j))^2 dx}{\int \mathcal{G}_i(x,\mu_i,\Sigma_i)^2 dx + \int \mathcal{G}_j(x,\mu_j,\Sigma_j)^2 dx} \tag{4.44}$$

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 $i_1, \cdots, i_k$ 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

$$\omega_i = \sum_{l=1}^{k} \omega_{i_l}, \quad \mu_i = \frac{\sum_{l=1}^{k} \omega_{i_l}\mu_{i_l}}{\omega_i} \tag{4.45}$$
$$P_i = \frac{\sum_{l=1}^{k} \omega_{i_l}(P_{i_l} + (\mu_{i_l} - \mu_i)(\mu_{i_l} - \mu_i)^T)}{\omega_i}.$$

4.3 Numerical Examples

In this section, the particle Gaussian mixture filter is applied to three test case problems to evaluate the filtering performance. Other nonlinear filters such as the UKF, PF and the Gaussian mixture 'blob' filter are also simulated for comparison. 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

$$E_{rms}(t) = \sqrt{\frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \|\hat{X}^j(t) - \hat{\mu}^j(t)\|_2^2}. \tag{4.46}$$

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

from $E_{rms}(t)$ as

$$\overline{E_{rms}} = \frac{1}{T}\sum_{t=1}^{T} E_{rms}(t). \tag{4.47}$$

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

$$\beta_{j,t} = (\hat{X}^j(t) - \hat{\mu}^j(t))^T (\hat{P}^j(t))^{-1} (\hat{X}^j(t) - \hat{\mu}^j(t)). \tag{4.48}$$

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

$$\beta_t = \frac{1}{N_{Mo}}\sum_{j=1}^{N_{Mo}} \beta_{j,t}. \tag{4.49}$$

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

$$\pi_n(x) = \sum_{i=1}^{M_n} \omega_i \mathcal{G}_i(x; \mu_i, P_i). \tag{4.50}$$

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 devel-

oped as part of the present work. Let the random vector $V_{\mathbb{1}}$ be defined as

$$V_{\mathbb{1}} = [\mathbb{1}_c^1(x) \cdots \mathbb{1}_c^i(x) \cdots \mathbb{1}_c^{M_n}(x)]^T, \tag{4.51}$$

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

$$E(V_{\mathbb{1}}) = [\omega_1 \cdots \omega_i \cdots \omega_{M_n}]^T. \tag{4.52}$$

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

$$\epsilon_v = V_{\mathbb{1}} - [\omega_1 \cdots \omega_i \cdots \omega_{M_n}]^T, \tag{4.53}$$

$$\epsilon_v^2 = \epsilon_v^T \epsilon_v. \tag{4.54}$$

Here the value of $V_{\mathbb{1}}$ 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

$$E(\epsilon_v^2) = \sum_{i=1}^{M_n} \omega_i(1-\omega_i), \tag{4.55}$$

$$E(\epsilon_v^2 - E(\epsilon_v^2))^2 = \sum_{i=1}^{M_n} \omega_i(1-\omega_i)((1-\omega_i)^3 + \omega_i^3) \tag{4.56}$$

$$+ \sum_k \sum_{\substack{j \\ j \neq k}} \omega_j \omega_k (\omega_j + \omega_k - 3\omega_j\omega_k) - (E(\epsilon_v^2))^2.$$

Let $V_{\mathbb{1}}^j(t)$ 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

$$\mathbb{1}_c^i(\hat{X}^j(t)) = \begin{cases} 1, & i = \arg\max \mathcal{G}_i(\hat{X}^j(t), \mu_i, P_i) \\ 0, & \text{otherwise} \end{cases} \tag{4.57}$$

Let the sum Sw_t be defined as

$$Sw_t = \sum_{i=1}^{N_{Mo}} \frac{\epsilon_v^j(t)^2 - E(\epsilon_v^j(t)^2)}{\sqrt{N_{Mo} E(\epsilon_v^j(t)^2 - E(\epsilon_v^j(t)^2))^2}} \tag{4.58}$$

The expectations involved in this sum are computed using the mixture weights $\omega_i^j(t)$ at time t during the run j. 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.,

$$\begin{aligned} &\beta_{j,t} = (\hat{X}^j(t) - \hat{\mu}_i^j(t))^T (\hat{P}_i^j(t))^{-1} (\hat{X}^j(t) - \hat{\mu}_i^j(t)), \\ &i = \arg\max N(\hat{X}^j(t), \mu_i^j(t), P_i^j(t)) \end{aligned} \tag{4.59}$$

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

$$L(t) = \frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \pi_t^j(\hat{X}^j(t)). \tag{4.60}$$

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

$$\hat{L} = \frac{1}{T} \sum_{t=1}^{T} L(t). \tag{4.61}$$

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

$$V^j\sigma_2(t) = \sum_{i=1}^{M_t} |2\Sigma_i|. \tag{4.62}$$

where M_t is the number of modes. Here $|.|$ 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

$$V\sigma_2(t) = \sum_{j=1}^{N_{Mo}} V^j\sigma_2(t). \tag{4.63}$$

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

$$\hat{V}\sigma_2 = \frac{1}{T}\sum_{i=1}^{T} V\sigma_2(t). \tag{4.64}$$

4.3.1 Example 1

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

$$x_{k+1} = \frac{x_k}{2} + \frac{25x_k}{(1+x_k^2)} + 8\cos(1.2k) + \nu_k. \tag{4.65}$$

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

$$z_k = \frac{x_k^2}{20} + n_k, \tag{4.66}$$

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[24, 23]. 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 4.1.

Table 4.1: UKF Parameter Values for PGM-I, test case 1. Reprinted with permission from[1, 2].

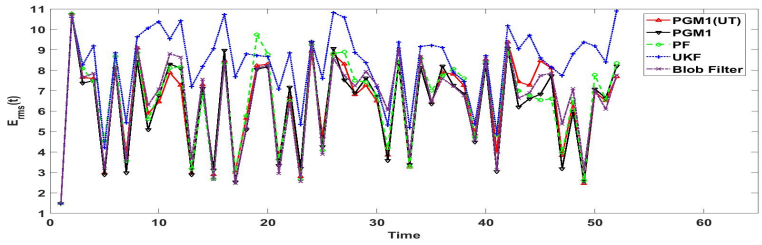
UKF parameters		
α	β	λ
1.3	1.5	0.2

The values of $E_{rms}(t)$ plotted in Figure 4.2a 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 Figure 4.2b show that the UKF and PF frequently oversteps the upper bound which marks inconsistent estimates. Furthermore, β_t computed using 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 Figures 4.2c and 4.2d show that only the blob Filter provides more informative estimates than the PGM-1 and PGM-1 UT.

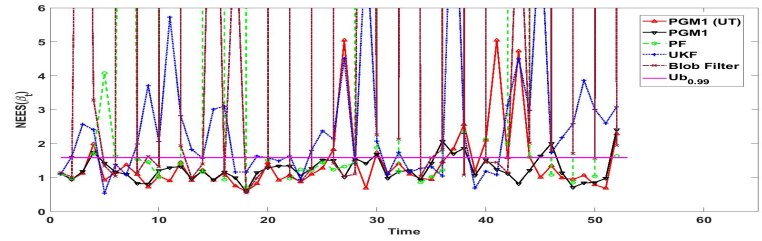
The time averaged values of RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for each filter are listed in Table 4.2. Also included is the fraction ($\beta_{c\%}$) of the time instants during which the computed averaged NEES result stayed within the 99 percent limits,i.e.,

$$\beta_{c\%} = \frac{\sum_{t=1}^{T} \mathbb{1}_{Ub0.99}(\beta_t)}{T} \tag{4.67}$$

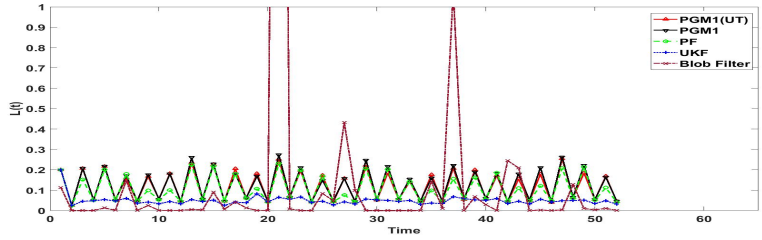
where $\mathbb{1}_{Ub0.99}(\beta_t)$ is the indicator function which equals 1 when $\beta_t < Ub_{0.99}$ and zero otherwise. The results presented in Table 4.2 clearly show that the PGM filter implementations offer accurate, consistent and informative estimates.



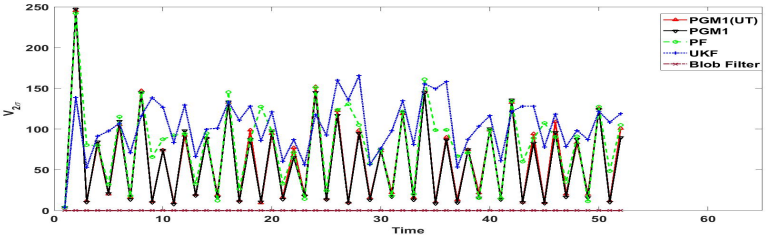
(a) $E_{rms}(t)$



(b) β_t



(c) $L(t)$



(d) $V_{2\sigma}$

Figure 4.2: PGM-I estimation results for bimodal one dimensional nonlinear model. Reprinted with permission from [1].

4.3.2 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

Table 4.2: Time averaged RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for PGM-I, test case 1. Reprinted with permission from [1].

Example 1: Results				
	$\overline{E_{rms}}$	$\beta_{t,c}(\%)$	$\hat{L}$	$\hat{V}\sigma_2$
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

Lorenz 63 system is described the the following set of equations,

$$\begin{aligned} \dot{x_1} &= \alpha(-x_1 + x_2), \;\; \alpha = 10 \\ \dot{x_2} &= \beta x_1 - x_2 - x_1 x_3, \;\; \beta = 28 \\ \dot{x_3} &= -\gamma x_3 + x_1 x_2 + \Gamma(t), \;\; \gamma = 8/3 \end{aligned} \tag{4.68}$$

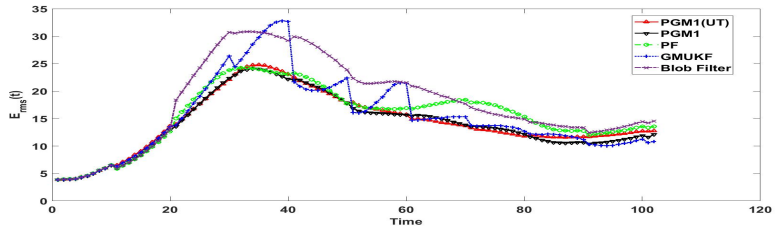
A scalar nonlinear measurement model(z_k) is considered which is given by

$$z_k = \sqrt{x_1(t)^2 + x_2(t)^2 + x_3(t)^2} + \nu_k. \tag{4.69}$$

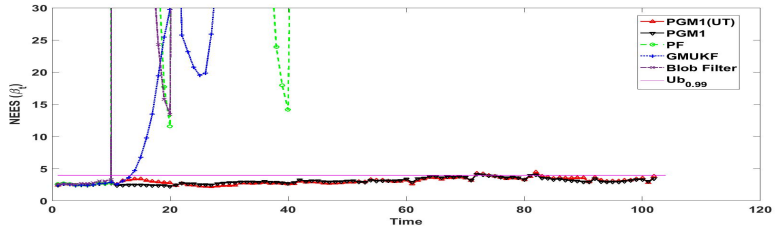
The process and measurement noise covariances are both set be equal to 1. The initial state of the system is characterized by the bimodal PDF

$$\begin{aligned} p_0(x) =& 0.9\mathcal{G}(x, [-0.2, -0.2, 8]^T, \sqrt{0.35} I_{3\times 3}) + \\ & 0.1\mathcal{G}(x, [0.2, 0.2, 8]^T, \sqrt{0.35} I_{3\times 3}) \end{aligned} \tag{4.70}$$

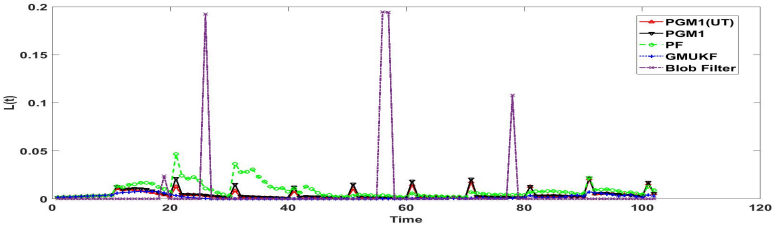
The state of the system is updated at a time step $\Delta t = 0.01s$. The measurements are recorded at the interval of ten time steps. This example has been considered previously in [20].



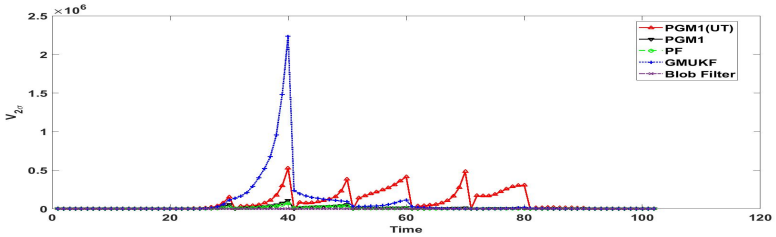
(a) $E_{rms}(t)$



(b) NEES(β_t)



(c) $L(t)$



(d) $V_{2\sigma}$

Figure 4.3: PGM-I estimation results for Lorenz 63 model. Reprinted with permission from [1].

The PGM1 filter, PGM1-UT filter, the PF, the blob filter and a conventional Gaussian mixture UKF [19] are employed in the estimation of the Lorenz63 system. The PGM filters and the SIR filter are implemented with 300 particles and M_{max} is set be 2. The UKF is implemented using the parameters listed in Table 4.1. 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 Figure 4.3a 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 Figure 4.3b. 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.

Table 4.3: Time averaged RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for PGM-I, test case 2. Reprinted with permission from [1].

Example 2: Results				
	$\overline{E_{rms}}$	$\beta_{t,c}(\%)$	$\hat{L}$	$\hat{V}\sigma_2(\times 10^4)$
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}

The averaged likelihoods($L(t)$) and $V\sigma_2$ volumes plotted in Figures 4.3c, 4.3d 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 4.3. The results clearly indicate that the PGM filters are more accurate and consistent than the PF, mixture UKF and blob filter.

4.3.3 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

$$\dot{x_i} = x_{i-1}(x_{i+1} - x_{i-2}) - x_i + F + \Gamma(t), \tag{4.71}$$

where $i = 1, 2, \cdots, 40$ [43]. 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,

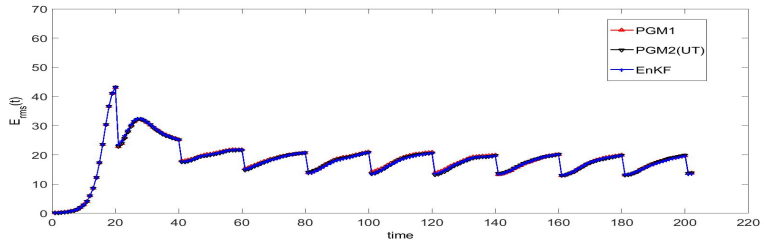
$$z_k = HX_k + \nu_k, \quad H_{i,j} = \begin{cases} 1, & j = 2i - 1 \\ 0, & \text{otherwise.} \end{cases}$$

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

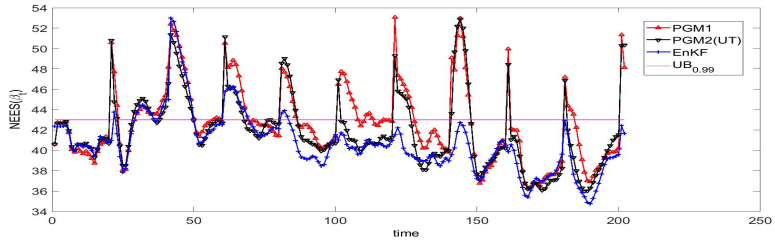
$$p_0(x) = \mathcal{G}(x, \mu_0, P_0), \tag{4.72}$$

where $\mu_0 = F\begin{bmatrix}1\cdots 1\cdots 1\end{bmatrix}^T, \mu_0 \in \mathbb{R}_{40\times 1}$ 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[31], 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 M_{max} is kept at 2. 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 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 Figure 4.4a, 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 Figure 4.4b. 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 Figures 4.4c, 4.4d 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 4.4 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.

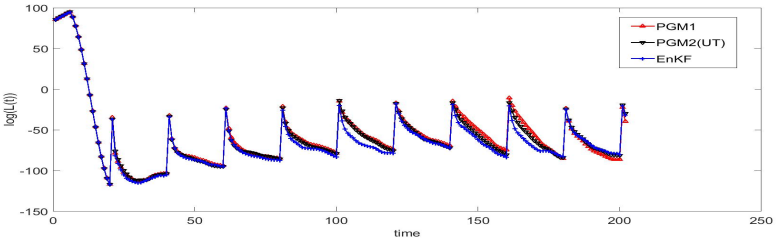
Our results in this chapter indicate that the PGM-I filter is capable of handling



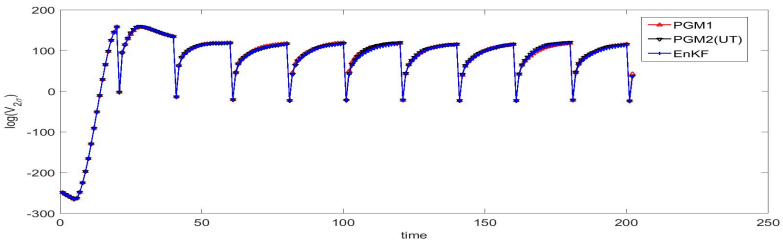
(a) $E_{rms}(t)$



(b) NEES(β_t)



(c) $log(L(t))$



(d) $log(V_{2\sigma})$

Figure 4.4: PGM-I estimation results for Lorenz96 system. Reprinted with permission from [1].

the general multimodal nonlinear filtering problems. The Monte Carlo uncertainty propagation method can handle the nonlinearity induced distortions and the splitting and merging of state PDF components during propagation step. However, since

Table 4.4: Time averaged RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for PGM-I, test case 3. Reprinted with permission from [1].

Example 3: Results				
	$\overline{E_{rms}}$	$\beta_{t,c}(\%)$	$\hat{logL}$	$\hat{logV}\sigma_2$
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

it relies on Kalman measurement update to obtain the posterior PDF, it cannot capture the splitting and merging of PDF components during measurement update. The design of a PGM filtering scheme that incorporates splitting of mixture modes during the measurement update is discussed in the next chapter.

5. PARTICLE GAUSSIAN MIXTURE FILTERS -II*

In chapter 4, we proposed a particle Gaussian mixture (PGM) filter for nonlinear estimation. The PGM-I filter uses the transition kernel of the state Markov chain to sample from the propagated prior. A Gaussian mixture representation of the propagated prior density is constructed by clustering the samples. The PGM-I algorithm incorporates the measurement data by updating individual mixture modes using the Kalman measurement update. The Kalman measurement update is inexact when the measurement function is nonlinear and leads to the restrictive assumption that the number of modes remain fixed during the measurement update. In this chapter, we introduce an alternate PGM-II filter that employs Markov Chain Monte Carlo (MCMC) sampling to perform the measurement update [4, 3]. The measurement update step in the Gaussian mixture filtering algorithm is discussed in section 5.1. A brief review of MCMC methods is given in section 5.2, following which the PGM-II filtering algorithm is introduced in section 5.3.

5.1 Gaussian Mixture Filtering: Measurement Update

In the measurement update step, the propagated PDF $\pi_n^-(x)$ is updated with the new measurement z_n according to the Bayes rule to obtain the posterior PDF $\pi_n(x)$.

$$\pi_n(x) = \frac{p(z_n/x)\pi_n^-(x)}{\int p(z_n/x')\pi_n^-(x')dx'}, \tag{5.1}$$

Let us assume that the propagated prior PDF $\pi_n^-(x)$ can be approximated by a

*Parts of this chapter were reprinted with permission from [3, 4].

weighted sum of Gaussian PDFs.

$$\pi_n^-(x) = \sum_{i=1}^{M^-(n)} \omega_i^-(n)\mathcal{G}_i^-(x;\mu_i^-(n),P_i^-(n)) \tag{5.2}$$

When the GMM representation of $\pi_n^-(x)$ is substituted in equations 1.3, 5.1, we get

$$\pi_n(x) = \frac{\sum_{i=1}^{M^-(n)} \omega_i^-(n)p(z_n/x)\mathcal{G}_i^-(x;\mu_i^-(n),P_i^-(n))}{\sum_{i=1}^{M^-(n)} \omega_i^-(n)\int p(z_n/x')\mathcal{G}_i^-(x';\mu_i^-(n),P_i^-(n))dx'}. \tag{5.3}$$

Let $l_i(n)$ be the likelihood that the measurement z_n came from the i^{th} mixture component.

$$l_i(n) \equiv \int p(z_n/x')\pi_{i,n}^-(x')dx'. \tag{5.4}$$

The expression for the posterior PDF given in equation 5.3 can be rewritten as follows.

$$\pi_n(x) = \sum_{i=1}^{M^-(n)} \underbrace{\frac{w_i^-(n)l_i(n)}{\sum_j w_j^-(n)l_j(n)}}_{w_i(n)} \underbrace{\frac{p(z_n/x)\pi_{i,n}^-(x)}{l_i(n)}}_{\pi_{i,n}(x)}. \tag{5.5}$$

This shows that the posterior PDF $\pi_n(x)$ can be represented as a mixture model $\{(\omega_i(n),\pi_{i,n}(x))\}, i \in \{1,\cdots,M^-(n)\}$ where,

$$\omega_i(n) = \frac{w_i^-(n)l_i(n)}{\sum_j w_j^-(n)l_j(n)}, \tag{5.6}$$

$$\pi_{i,n}(x) = \frac{p(z_n/x)\pi_{i,n}^-(x)}{l_i(n)}. \tag{5.7}$$

This mixture model has $M^-(n)$ components and the mixands $\pi_{i,n}(x)$ are not guar-

anteed to be Gaussian when the measurement function is nonlinear. The Kalman measurement update used in PGM-I filter obtains unimodal Gaussian approximations of individual mixands $\pi_{i,n}(x)$ by linearizing the measurement model. The result is an $M^{-}(n)$ component GMM approximation of the posterior PDF. However, when the measurement function is highly nonlinear or ambiguous, a single Gaussian component from the predicted prior can give rise to multiple posterior modes.

5.2 Markov Chain Monte Carlo

The MCMC methods are a class of algorithms that are used to generate samples from probability distributions that are not amenable to direct sampling [44]. In the present chapter we consider the Metropolis Hastings algorithm which relies on a proposal distribution to generate the samples [45, 46]. A sequence of candidate points are drawn from the proposal distribution. The candidate points are then retained or discarded based on an acceptance probability computed from the target distribution. Interestingly, computing the acceptance probability requires only a function that is proportional to the target distribution. This is very useful in cases where computing the normalization constant for the target distribution is difficult. Formally, let $P(X)$ be the target distribution from which the samples are to be generated. Let $Q(X^i|X^{i-1})$ be the proposal distribution. Then the MCMC algorithm proceeds as follows. Let X^{t-1} be the sampled state at $t-1$. Then generate $X^{t*} \sim Q(X|X^{t-1})$. The candidate state X^{t*} is then chosen or not based on the acceptance probability α. The acceptance probability is computed as

$$\alpha = min\{1, \frac{Q(X^{t-1}|X^{t*})P(X^{t*})}{Q(X^{t*}|X^{t-1})P(X^{t-1})}\} \tag{5.8}$$

When the proposal distribution is symmetric, i.e., $Q(X^{t-1}|X^{t*}) = Q(X^{t*}|X^{t-1})$, the above expression simplifies to

$$\alpha = min\{1, \frac{P(X^{t*})}{P(X^{t-1})}\} \tag{5.9}$$

Gaussian proposal densities of the form $Q(X^{t*}|X^{t-1}) = \mathcal{N}(X, X^{t-1}, \Sigma)$ are symmetric. It can be shown that the sampling rule given above is constructed so that the target distribution $P(X)$ is the equilibrium distribution of the resulting Markov chain. This implies that the initial samples may not be distributed according to $P(X)$. As a result, all points sampled during an initial burn-in period T_{br} are discarded.

In theory, the MH algorithm is capable of generating samples from complex multimodal distributions. However, generating a representative sample from a multimodal distribution may require a long burn-in period and a large sample size. Consider the case when the target distribution is multimodal with well separated modes. Assume that the proposal distribution is Gaussian $\mathcal{N}(X, X^{t-1}, \Sigma)$ as is often the case. If Σ is too small compared to the modal covariances, then the probability that the chain will jump from one target mode to another will be quite low over a finite time period. On the other hand, choosing a large Σ will result in too many jumps to regions that are of no interest. This is a major shortcoming of the MH algorithm in sampling from multimodal posterior distributions. Parallellizable MCMC algorithms that split the state space into partitions and allow asynchronous sampling from individual partition elements have been proposed recently [47]. In this work, we propose a similar approach to sample from multimodal posterior PDFs. As mentioned before, the MH algorithm generates a Markov Chain whose steady state distribution will be the target PDF $P(X)$. Hence the samples drawn by MH over a finite time are not from

the exact target distribution. Perfect sampling algorithms termed coupling from the past (CFTP), that can draw samples from the exact target distribution have also garnered significant attention recently [48].

5.3 PGM-II Filter

In this section we present a step by step description of the proposed PGM-II filter and an associated convergence result.

5.3.1 The PGM-II algorithm

The PGM-II filter relies on a Gaussian mixture representation of the state PDF. However, unlike the PGM-I filter, it is not essential for the operation of PGM-II algorithm that we obtain a functional representation of the posterior PDF.

Assumption 6. *The predicted prior PDF and the filtered PDF can be represented as a GMM.*

Given an ensemble of states $S_{n-1} = \{x_{n-1}^1 \cdots x_{n-1}^N\}$ from the prior PDF at time $n-1$, the PGM-II Filtering algorithm is composed of the three basic steps described below.

1. **Prediction**: During the prediction step, the PGM-II filter generates an ensemble S_n^- from the predicted prior $\pi_n^-(x)$ using samples drawn from the prior PDF $\pi_{n-1}(x)$, i.e., the ensemble S_{n-1}, and the Markov transition kernel $P(x'/x)$. A pictorial representation of the prediction is given as the first step in Figure 5.1.

2. **Clustering** : A functional representation of $\pi_n^-(x)$ in the form of a GMM is recovered from the ensemble S_n^- using a clustering scheme $\mathcal{C}$ [33, 34]. The output of the clustering scheme is composed of the mixture weights $\omega_i^-(n)$, means $\mu_i^-(n)$ and covariances $P_i^-(n)$. The ellipsoids obtained at the end of clustering step in Figure 5.1

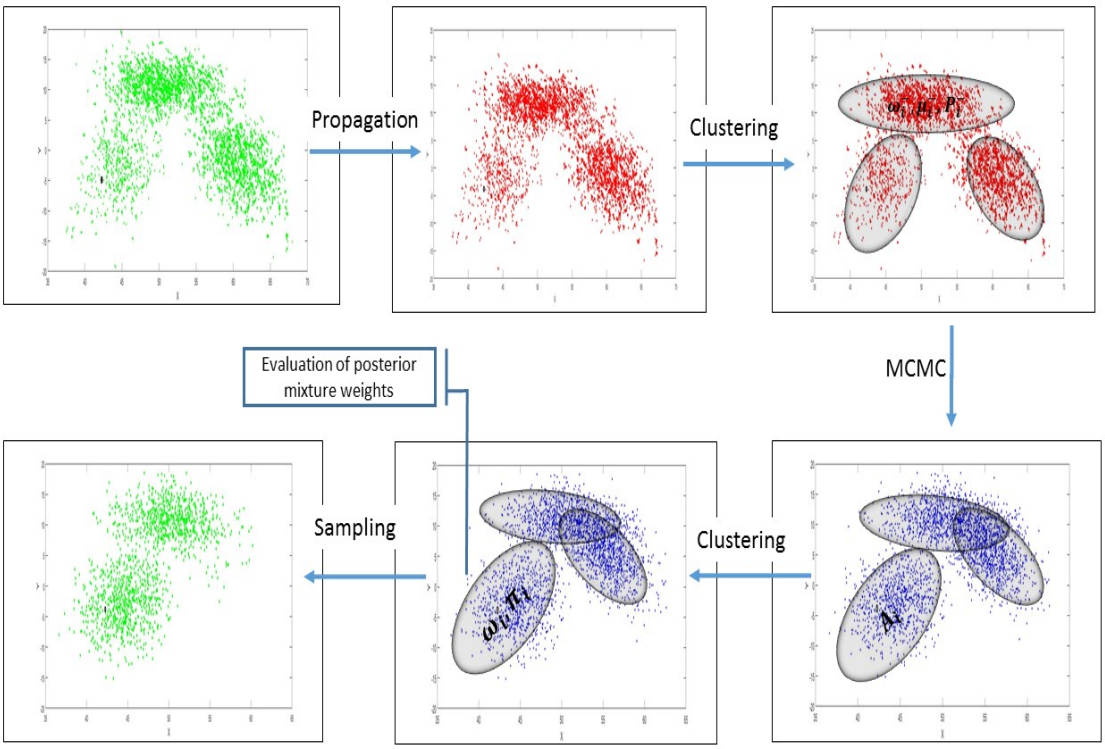


Figure 5.1: PGM-II Filter-Prediction and Update. Reprinted with permission from [3, 4].

represent the Gaussian mixture components. In particular,

$$\pi_n^-(x) = \sum_{i=1}^{M^-(n)} \omega_i^-(n)\mathcal{G}_i(x; \mu_i^-(n), P_i^-(n)) \tag{5.10}$$

3. **Measurement update**: The PGM-II filter relies on a parallellized MCMC method to perform the measurement update. The parallellized MCMC update is broken down into the following four steps.

(a) Sample from the i^{th} posterior mixture component $\pi_{i,n}(x)$ (equation 5.7) using MCMC to obtain the i^{th} posterior component ensemble A_i.

(b) Cluster the i^{th} posterior component sample A_i to obtain a functional representation for the component PDF $\pi_{i,n}(x)$. r mixture component weight $w_i(n)$ (equation 5.6).

(c) Evaluate the i^{th} posterior mixture component weight $w_i(n)$ (Eq. 5.6).

(d) Sample from the mixture model $\{w_i, \pi_{i,n}(x)\}$ to obtain a full posterior ensemble S_n.

The four step update process is described in more detail below.

Let $p_n(z_n/x)$ be the measurement likelihood. Then the posterior distribution is proportional to the product of the predicted prior and the likelihood $p_n(z_n/x)$.i.e,

$$\pi_n(x) \propto p_n(z_n/x)\pi_n^-(x). \tag{5.11}$$

When the predicted prior is represented by a GMM, the posterior is given by equation 5.3 We rewrite the posterior PDF in its mixture form as obtained in equation 5.5:

$$\pi_n(x) = \sum_{i=1}^{M^-(n)} w_i(n)\pi_{i,n}(x). \tag{5.12}$$

Furthermore, from equation 5.7,

$$\pi_{i,n}(x) \propto p_n(z_n/x)\pi_{i,n}^-(x). \tag{5.13}$$

In step 3a of the measurement update, the PGM-II filter generates ensembles A_i from the mixture components $\pi_{i,n}(x), i \in \{1, 2, \cdots M^-(n)\}$ using MCMC since $p_n(z_n/x)$ is given and $\pi_{i,n}^-(x)$ is known from the clustering step. From a computational standpoint, it is much more appealing to perform MCMC sampling on the individual mixture components $\pi_{i,n}(x)$ as opposed to the full posterior PDF $\pi_n(x)$. This com-

pletes step 3a. Due to the random walk behavior of MCMC, consecutive samples from A_i will be correlated. To remove correlations, we propose clustering the samples and obtaining a functional representation for the underlying component PDF $\pi_{i,n}(x)$. Notice that the mixture representation of $\pi_{i,n}(x)$ will be parameterized by expectations of various functions of the component random variable. The ergodicity of the chain will ensure that sample averages computed from MCMC samples during clustering will converge to these expectations, in spite of the correlations. Once a mixture representation for $\pi_{i,n}(x)$ is constructed, we can obtain independent samples from it by direct sampling. The clustering of A_i to obtain functional representation of $\pi_{i,n}(x)$ completes the step 3b of measurement update.

Notice that in equation 5.12, each component PDF $\pi_{i,n}(x)$ has a mixing probability $w_i(n)$ associated with it. Step 3c consists of obtaining these weights. However, to compute the mixture weights, we need to evaluate the modal likelihoods $l_i(n)$, given by the integral in equation 5.4. Evaluating this integral is non trivial when the measurement function is nonlinear. So an approximation is used in the computation of $w_i(n)$. The calculation of approximate modal likelihoods is discussed in detail in section 5.4.

From the component PDFs $\pi_{i,n}(x)$ in step 3b and the weights $w_i(n)$ in step 3c, we can obtain a mixture representation of the posterior PDF as given in equation 5.12. Given the mixture representation, a sample X from the full posterior PDF $\pi_n(x)$ can be obtained via the two step approach given below.

(i) Choose a component by sampling k from $\{1, 2, \cdots, M^-(n)\}$ with probability $w_k(n)$.

(ii) Draw a sample X from the component PDF $\pi_{k,n}(x)$.

This is the sampling process given in step 3d which completes the measurement

update step. This is also the last step in Figure 5.1, at the end of which we obtain an ensemble S_n from the full posterior, which is then propagated till the next step and the process repeats.

Algorithm 4 gives a pseudo code description of the PGM-II filter. More details on how the MCMC method is implemented to generate the ensembles A_i is presented in section 5.4

Algorithm 4 PGM-II Algorithm

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

(a) Sample N particles $X^{(i)}$ from π_{n-1} and the transition kernel $p_n(x'/x)$ as follows:
   i. Sample $X^{(i)}$ from $\pi_{n-1}(.)$.
   ii. Sample $X'^{(i)}$ from $p(./X^{(i)})$.

(b) 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 $\{w_i^-(n), \mu_i^-(n), P_i^-(n)\}$.

(c) Use MCMC to sample from the component posteriors $\pi_{i,n}(x)$ to generate the ensembles A_i

(d) Compute the mixture weights $w_i(n)$ by evaluating the sequence of modal likelihoods $l_i(n)$ using equations 5.4, 5.6

(e) Sample N particles from the weighted collection of ensembles $\{(w_i(n), A_{n,i})\}$

(f) n = n+1, go to Step 1.

5.3.2 Analysis of the PGM-II algorithm

In the following, we prove that the PGM-II filter density converges in probability to the true filter density under certain assumptions.

We showed in the previous chapter and in [49] that under the condition of exponential forgetting of initial conditions, the true filter density can be approximated

arbitrarily well with arbitrarily high confidence given that the sampling error in each step is small. We establish a similar result in the following.

Define:

$$P(\hat{\pi}_{n-1}) \equiv \hat{\pi}_n^- = \sum_{i=1}^{M^-(n)} \hat{\omega}_i^-(n)\mathcal{G}_i(x; \hat{\mu}_i^-(n), \hat{P}_i^-(n)), \tag{5.14}$$

$$\hat{P}(\hat{\pi}_{n-1}) \equiv \hat{\hat{\pi}}_n^- = \sum_{i=1}^{M^-(n)} \hat{\hat{\omega}}_i^-(n)\mathcal{G}(x; \hat{\hat{\mu}}_i^-(n), \hat{\hat{P}}_i^-(n)), \tag{5.15}$$

$$F_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\omega}_i(n)\mathcal{G}(x; \hat{\mu}_i(n), \hat{P}_i(n)), \tag{5.16}$$

$$\hat{F}_{z_n}(\hat{\pi}_{n-1}) = \sum_{i=1}^{M(n)} \hat{\hat{\omega}}_i(n)\mathcal{G}(x; \hat{\hat{\mu}}_i(n), \hat{\hat{P}}_i(n)). \tag{5.17}$$

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 6. *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:*

$$Prob(|\hat{\hat{\omega}}_i^-(n) - \hat{\omega}_i^-(n)| > \epsilon') < \delta', \tag{5.18}$$

$$Prob(|\hat{\hat{\mu}}_i^{j-}(n) - \hat{\mu}_i^{j-}(n)| > \epsilon') < \delta', \tag{5.19}$$

$$Prob(|\hat{\hat{P}}_i^{jk-}(n) - \hat{P}_i^{jk-}(n)| > \epsilon') < \delta', \tag{5.20}$$

for all i, j, k, where $\hat{\mu}_i^{j-}$ represents the j^{th} element of the mean vector $\hat{\mu}_i^-$ and $\hat{P}_i^{jk-}$ represents the $(j,k)^{th}$ element of the covariance matrix $\hat{P}_i^-$.

Lemma 7. *Let* $|\hat{\hat{\omega}}_i^-(n) - \hat{\omega}_i^-(n)| < \epsilon'$, $|\hat{\hat{\mu}}_i^{j-}(n) - \hat{\mu}_i^{j-}(n)| < \epsilon'$, *and* $|\hat{\hat{P}}_i^{jk-}(n) - \hat{P}_i^{jk-}(n)| < \epsilon$ *for all* i, j, k. *Then , given that the state of the system* $x \in \Re^d$, *there exists* $C^-(n) < \infty$ *such that* $||\hat{\hat{\pi}}_n^- - \hat{\pi}_n^-|| < C^-(n)d\epsilon'$.

Lemma 6 and 7 are proved in [49].

Lemma 8. *Let,* $||\hat{\hat{\pi}}_n^- - \hat{\pi}_n^-|| < \epsilon^-$, *then given the posterior* $\hat{\hat{\pi}}_n^* = F_{z_n}(\hat{\pi}_{n-1})$, *there exists* $k(n) < \infty$ *s.t:* $||\hat{\hat{\pi}}_n^* - \hat{\pi}_n|| < k(n)\epsilon^-$.

Proof. Let

$$P_1 = \int p_n(z_n/x')\hat{\pi}_n^-(x')dx', \tag{5.21}$$

$$P_2 = \int p_n(z_n/x')\hat{\hat{\pi}}_n^-(x')dx'. \tag{5.22}$$

Then

$$\begin{aligned}
||\hat{\hat{\pi}}_n^* - \hat{\pi}_n|| &= \int |(\frac{\hat{\pi}_n^-(x')}{P_1} - \frac{\hat{\hat{\pi}}_n^-(x')}{P_2})p_n(z_n/x')|dx' \\
&= \int |(\frac{\hat{\pi}_n^-(x') - \hat{\hat{\pi}}_n^-(x') + \hat{\hat{\pi}}_n^-(x')}{P_1} - \frac{\hat{\hat{\pi}}_n^-(x')}{P_2})p_n(z_n/x')|dx' \\
&\leq \int |\frac{P_2 - P_1}{P_1 P_2}|\hat{\hat{\pi}}_n^-(x')p_n(z_n/x')dx' + \\
&\quad \int |\hat{\pi}_n^-(x') - \hat{\hat{\pi}}_n^-(x')|\frac{p_n(z_n/x')}{P_1}dx' \\
&= |\frac{P_2 - P_1}{P_1}| + \max_{x'} \frac{p_n(z_n/x')\delta^-}{P_1} \\
&= \frac{|\int(\hat{\pi}_n^-(x') - \hat{\hat{\pi}}_n^-(x'))p_n(z_n/x')dx'|}{P_1} + \max_{x'} \frac{p_n(z_n/x')\epsilon^-}{P_1} \\
&\leq 2\max_{x'} \frac{p_n(z_n/x')\epsilon^-}{P_1}
\end{aligned} \tag{5.23}$$

Choosing $k(n) = 2\max_{x'} \frac{p_n(z_n/x')}{P_1}$ completes the proof. □

Let $\hat{\hat{\pi}}_n^*$ be the exact posterior evaluated from the propagated PDF $\hat{\hat{\pi}}_n^-$. The filtered

PDF $\hat{\hat{\pi}}_n$ is a GMM representation of $\hat{\hat{\pi}}^*_n$. By Lemma 6, there exists an upperbound on the number of samples $N^*_{\epsilon^{'},\delta'}$ such that the mixture parameters of $\hat{\hat{\pi}}^*_n$ are estimated with an accuracy of $\epsilon^{'}$ with a confidence $1-\delta^{'}$ if the MCMC draws these many samples. Let the number of particles used in PGM-II filter be $N = \max(N_{\epsilon^{'},\delta'}, N^*_{\epsilon^{'},\delta'})$. Therefore:

$$||\hat{\pi}_n - \hat{\hat{\pi}}_n|| = ||\hat{\pi}_n - \hat{\hat{\pi}}^*_n + \hat{\hat{\pi}}^*_n - \hat{\hat{\pi}}_n||, \tag{5.24}$$

$$\leq ||\hat{\pi}_n - \hat{\hat{\pi}}^*_n|| + ||\hat{\hat{\pi}}^*_n - \hat{\hat{\pi}}_n|| \tag{5.25}$$

From Lemma 6, 7 and 8, we have

$$Prob(||\hat{\pi}_n - \hat{\hat{\pi}}^*_n|| > k(n)C^{-}(n)d\epsilon') < \delta' \tag{5.26}$$

From Lemma 6 and 7, we also have

$$Prob(||\hat{\hat{\pi}}^*_n - \hat{\hat{\pi}}_n||) > C(n)d\epsilon^{'}) < \delta^{'} \tag{5.27}$$

Clearly,

$$Prob\Big(||\hat{\pi}_n - \hat{\hat{\pi}}_n|| > \big(k(n)C^{-}(n) + C(n)\big)d\epsilon^{'}\Big) < 2\delta' \tag{5.28}$$

Hence, by choosing ϵ' such that $\epsilon = \big(k(n)C^{-}(n)+C(n)\big)d\epsilon^{'}$, and δ' such that $\delta = 2\delta'$, and $N = \max(N_{\epsilon',\delta'}, N^*_{\epsilon',\delta'})$, we get

$$Prob(||\hat{\pi}_n - \hat{\hat{\pi}}_n|| > \epsilon) < \delta. \tag{5.29}$$

This proves that if the number of samples used to approximate the predicted and posterior GMM parameters are more than N, then the sampling error stays within

the desired bounds with confidence $1 - \delta$. Assuming that the underlying Markov chain has the exponential forgetting property, this suffices to show the convergence in probability of the PGM-II density to the true filter density as proved by Lemma 2 in chapter 4.

5.3.3 Relationship With Other Nonlinear Filters

As mentioned earlier, the PGM-II filter inherits its particle based approach to uncertainty propagation from the PGM-I filter. This is a feature shared also by sequential importance resampling (SIR) type particle filters. Given the process model of the system, samples from the propagated PDF can be obtained in a straightforward fashion using the underlying Markov transition kernel. Furthermore the particle uncertainty propagation does not require any linearizing approximations on the process model or the state PDF. This stands in contrast with mixture filters such as the Gaussian mixture EKF/UKF [19]. These filters were proposed to incorporate the multimodality of the state PDF by using a bank of parallel nonlinear Kalman filters. The Gaussian mixture EKF/UKF linearizes the process model separately within the support of each mixture mode. As a result, these filters could not incorporate the splitting and coalescing of individual Gaussian components resulting from nonlinear uncertainty propagation. Several approaches have been proposed in recent years to overcome the shortcomings of the Gaussian mixture EKF/UKF. The Gaussian mixture 'blob' filter, that has been proposed recently, attempts to circumvent the effects of nonlinear uncertainty propagation by enforcing LMI based upper bounds on the component covariances [22]. The adaptive Gaussian sum filter linearizes the process model but attempts to minimize the approximation errors by adjusting the mixture weights during the propagation stage [20]. Adaptive entropy-based Gaussian-mixture information synthesis (AEGIS) is another approach that splits mixture modes based

on entropy considerations [21]. These approaches require frequent optimization or entropy calculations to perform the weight adjustment/splitting calculations. Furthermore, they rely on linearized models between the weight adjustments/component splits. The PGM-II filter inherits the relatively inexpensive particle based propagation algorithm used in PGM-I filters. It obtains the GMM parameters of the propagated PDF using a clustering algorithm. As a result, it can compute the number of mixture modes online during the estimation based on the distribution of the samples in state space.

Unlike the PGM filters, the PF does not obtain a Gaussian mixture representation of the state PDF. Instead, it relies on importance sampling to obtain particles from the posterior PDF. In basic implementations such as the SIR, the importance density is the same as the propagated prior. The particles drawn from the importance density are assigned weights to account for the disparity with true posterior. The main drawback of the PF is the increasing difficulty to sample particles from regions of high likelihood as the number of dimensions is increased. This problem is known as particle depletion. The number of particles required to prevent depletion increases exponentially with the dimensions of state space. Hence the PF is prone to the curse of dimensionality [25]. Several approaches have been proposed to reduce the risk of particle depletion. Filters such as the unscented particle filter attempt to obtain better importance densities so that particles are sampled from the right regions of the state space[50]. Homotopy methods are another class of approaches that attempt to create a flow of the density that results in the posterior PDF by solving a partial differential equation [51]. A mixture filter equivalent of particle filters that rely on Gaussian kernels in the place of particles has been proposed recently. This approach known as blob filter samples Gaussian kernels from the prior PDF like particles [22]. The finite covariance of the Gaussian kernels make them less

prone to the problem of depletion. The blob filter uses an EKF approximation for propagation and uses a combination of local linearization and resampling to perform the measurement update. It is hence a Gaussian mixture filter with relatively large number of components, each having a small covariance.

The PGM-I filter is free from the problem of particle depletion since it relies on a Kalman measurement update of the Gaussian mixture propagated PDF. However, the linearization involved in Kalman measurement update cannot incorporate the splitting and coalescing of the mixture components during the update step. The PGM-II filter is proposed to incorporate these features of multimodal nonlinear filtering by replacing the Kalman measurement update with a parallelized MCMC algorithm. Apart from the Gaussian mixture representation of the propagated PDF, it does not make any restrictive assumptions on the state space description or the nature of uncertainties. The parallelized MCMC approach adopted in PGM-II is specifically designed to sample from multimodal posterior PDFs.

5.4 Implementation

In this section, we discuss certain aspects concerning the implementation of the PGM-II filter in greater detail. To sample from the posterior PDF $\pi_n(x)$ using MCMC, we need a function that is at least proportional to it. It can be seen from equation 5.11 that constructing such a function is straightforward once we have a functional form for the predicted prior $\pi_n^-(x)$. The PGM-II filter relies on a GMM representation of the predicted prior obtained by clustering the predicted ensemble. Hence it is important that the clustering scheme is able to provide a GMM that closely models the true distribution of the predicted ensemble. In the present work, we have used an approach that relies on k-means clustering algorithm to obtain the GMM parameters. The simple k-means clustering algorithm requires the number

of mixture components to be input externally. To overcome this limitation, we have developed a clustering scheme which determines the optimal number of clusters given an upper bound on this number. This clustering scheme relies on the likelihood agreement measure (LAM) to arrive at the optimal number of mixture modes [21]. Let $S_{a,L} = \{\omega_{a,i}, \mu_{a,i}, P_{a,i}\}$ be an L component GMM. Then the LAM of the model $S_{a,L}$ in describing the data set $A = \{x_k\}$ is computed as

$$LAM_{S^{a,L}} = \sum_{k=1}^{N} \sum_{i=1}^{L} \omega_{a,i} \mathcal{G}_i^-(x_k; \mu_{a,i}, P_{a,i}) \tag{5.30}$$

The modified K-means clustering algorithm that was described in the previous chapter is employed in PGM-II filter as well.

Once a GMM representing the predicted prior PDF is obtained (equation 5.3), we have a functional form for the propagated PDF. Then if the posterior PDF is given by equation 5.12, we have

$$\pi_{i,n}(x) \propto \mathcal{G}_i^-(x; \mu_i^-(n), P_i^-(n)) p(z_n/x). \tag{5.31}$$

This provides a known function that is proportional to the component posterior $\pi_{i,n}(x)$ which can be used with an MCMC algorithm to draw samples from $\pi_{i,n}(x)$. The proposal distribution $Q_i(X^i|X^{i-1})$ obtains the candidate samples for the MCMC algorithm. In the present work we have used Gaussian proposals of with the functional form $\mathcal{G}(X^i, X^{i-1}, K_p\Sigma)$, where K_p is a positive constant. The covariance Σ can be chosen as the component covariance of the predicted prior $P_i^-(n)$. Covariance of the approximate posterior obtained by updating $\pi_n^-(x)$ with EKF/UKF can be another option. Choosing a proposal covariance that is too large or too small can delay the convergence of the Markov chain. The first candidate point X^0 for starting

the MCMC simulation can be sampled from the approximate component posterior obtained by updating $\pi_n^-(x)$ with EKF/UKF. The mean of this approximate component posterior can also be used as X^0.

There are several ways in which the modal likelihood $l_i(n)$ (equation 5.4) can be approximated,

1. **Importance Sampling**: Consider the measurement PDF $P_Z(z)$ given by

$$P_Z(z) = \int p_n(z/x')\mathcal{G}_i^-(x; \mu_i^-(n), P_i^-(n))dx'. \tag{5.32}$$

The integral in equation 5.32 can be estimated using the importance sampling technique as follows. A set of particles $\{x_k\}$ are drawn from an importance density $Q(X)$ from which direct sampling is possible. Then an estimator for $l_i(n)$ can be constructed as

$$l_i(n) \approx \frac{1}{M}\sum_{k=1}^{M} \frac{p_n(z/x_k)\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))}{Q(x_k)} \tag{5.33}$$

In order for the above estimator to be accurate, the importance density must have certain characteristics such as $Q(x_k) > 0$ whenever the product of the densities $p_n(z/x_k)$ and $\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))$ is nonzero. The choice of the importance density $Q(X)$ has to be exercised carefully and a good choice is a density that is proportional to $p_n(z/x_k)\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))$. However, directly sampling from such a density may not be feasible. Instead we choose a density that closely resembles the product and readily available such as

(a) Component density from the predicted prior: In this case

$$Q(X) = \mathcal{G}_i^-(x; \mu_i^-(n), P_i^-(n)), \tag{5.34}$$

as sampling from a Gaussian density is straightforward. Consequently, the expression 5.33 for the estimator reduces to

$$l_i(n) \approx \frac{1}{M} \sum_{k=1}^{M} p_n(z/x_k) \tag{5.35}$$

(b) Component density from the approximate posterior: We have

$$\pi_{i,n}(x) \propto p_n(z/x_k)\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n)). \tag{5.36}$$

The posterior $\pi_{i,n}(x)$ can be approximated by a Gaussian PDF by performing a EKF/UKF measurement update on $\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))$. This approximate posterior can be chosen as the importance density $Q(x)$.

2. **From the MCMC samples**: As mentioned previously, the PGM-II filter generates an ensemble A_i from the posterior mixture component $\pi_{i,n}(x)$ using MCMC sampling. Let $\eta(x)$ be a proper PDF. Then

$$\int_{\mathcal{R}^n} \eta(x)dx = 1 \tag{5.37}$$

Multiplying the integrand in the numerator and denominator by $\pi_{i,n}(x)$ and substituting from equation 5.7

$$\begin{aligned} &\int_{\mathcal{R}^n} \frac{\eta(x)\pi_{i,n}(x)l_i(n)}{p_n(z/x_k)\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))}dx = 1 \\ \Rightarrow &\int_{\mathcal{R}^n} \frac{\eta(x)\pi_{i,n}(x)}{p_n(z/x_k)\mathcal{G}_i^-(x_k; \mu_i^-(n), P_i^-(n))}dx = \frac{1}{l_i(n)} \end{aligned} \tag{5.38}$$

Since A_i are samples from $\pi_{i,n}(x)$, an importance sampling approximation to the

above integral can be arrived at as:

$$\frac{1}{l_i(n)} \approx \sum_{i=1}^{N_{i,n}} \frac{\eta(x_i)}{p_n(z/x_i)\mathcal{G}_i^-(x_i;\mu_i^-(n),P_i^-(n))}. \tag{5.39}$$

Hence an estimate of $l_i(n)$ can be computed by evaluating the sum given in equation 5.39 using the MCMC samples and taking the reciprocal [52]. The density $\eta(x)$ can be chosen as the approximate posterior obtained through EKF/UKF update. Another option is to use a Gaussian PDF parameterized by the mean and covariance of the MCMC samples. In the present work, we have adopted the MCMC approach to compute $l_i(n)$.

5.5 Numerical Examples

In this section, we employ the PGM-II Filter in the estimation of two test case systems to study the filtering performance. The results are compared with that of other nonlinear filters such as UKF, PF, PGM-I filter and blob filter. A basic sequential importance resampling (SIR) implementation of the PF is considered. The PGM-I variant which uses the unscented transform to perform the measurement update, i.e., the PGM-I(UT) filter, is used in this comparison study [2]. The estimation results are compared for their accuracy, consistency and informativeness. The accuracy of estimates is evaluated in terms of a Monte Carlo averaged root mean squared error ($E_{rms}(t)$).The value of $E_{rms}(t)$ is computed as

$$E_{rms}(t) = \sqrt{\frac{1}{N_{Mo}} \sum_{j=1}^{N_{Mo}} \|x_{j,t} - \mu_{j,t}\|_2^2}, \tag{5.40}$$

where $x_{j,t}$ and $\mu_{j,t}$ represent the actual and estimated states at the time instant t during the jth Monte Carlo run. Also evaluated is the time averaged error ($\overline{E_{rms}}$)

given by

$$\overline{E_{rms}} = \frac{1}{T}\sum_{t=1}^{T} E_{rms}(t). \tag{5.41}$$

The NEES test is employed to evaluate the consistency of the filtered PDF. The NEES test statistic ($\beta_{j,t}$) for a unimodal Gaussian PDF is given by

$$\beta_{j,t} = (x_{j,t} - \mu_{j,t})^T \mathbf{P_{j,t}^{-1}} (\mathbf{x_{j,t}} - \mu_{\mathbf{j,t}}), \tag{5.42}$$

where $P_{j,t}$ represents the covariance of the 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

$$\beta_t = \frac{1}{N_{Mo}}\sum_{j=1}^{N_{Mo}} \beta_{j,t}. \tag{5.43}$$

When $x \in \mathbb{R}^n$ is distributed normally, the statistic given by $N_{Mo}\beta_t$ is distributed according to a χ^2 distribution with nN_{Mo} degrees of freedom. This allows us to test for the consistency of the estimates by checking whether the value of the test statistic falls within probable bounds of the corresponding χ^2 random variable.

The informativeness of estimates is tested by comparing the volume of state space that contains a fixed fraction of the total probability. When the state PDF is Gaussian, the fraction of probability f_p contained in an $n - sigma(n \in Z^+)$ ellipse is only a function of the dimension d of the state space, i.e., $f_p = f_p(n, d)$. In this work, we compare the informativeness of the estimates in terms of the volume of state space $V\sigma_2$ that contains the fraction $f_p = f_p(2, d)$ of total probability. When the state PDF is represented by a GMM, this volume can be computed as the sum of the $2 - sigma$ volumes of individual mixture modes,i.e.,

$$V\sigma_2 = \sum_{i=1}^{L} |2\Sigma_i|. \tag{5.44}$$

5.5.1 Example 1

In test case 1, we consider a variant of the well known one-dimensional estimation problem [23, 24]

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos[1.2(k-1)] + w_{k-1}. \tag{5.45}$$

In the absence of noise and the cosine forcing term, the process model has three equilibrium points at $x = \pm 7, 0$. However, only the nonzero equilibrium points are stable. We define a multimodal measurement function

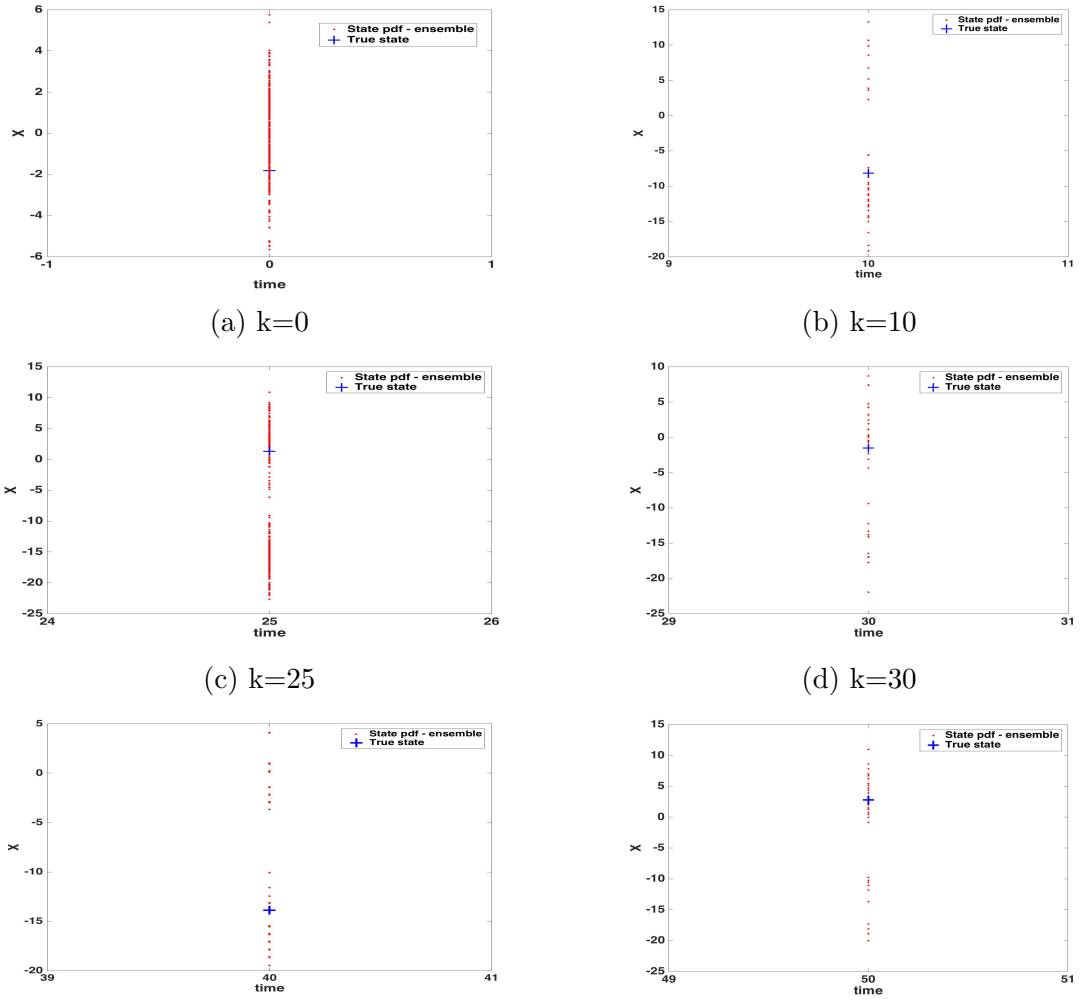
$$z_k = 4\sin(8x_k) + \nu_k. \tag{5.46}$$

Clearly, one can find several roots for the equation $g(x_k) = z_k$ given any measurement z_k. The coefficient of x_k, taken as 8 in this example, can be adjusted to increase the multimodality of the measurement model. The process and measurement noises are assumed to be independent zero mean Gaussian random variables with covariances Q=6, R=0.1 respectively. Measurements are recorded at every other instant. The evolution of the multimodal filtered PDF is presented in Figure 5.2.

The estimation is performed for a duration of 50 time steps and repeated over 50 Monte Carlo runs. The PF is implemented as an SIR with 80 particles. The UKF parameters are listed in Table 5.1. The PGM-II filter and the PGM-I filter are

Table 5.1: UKF Parameter Values for PGM-II, test case 1. Reprinted with permission from [3, 4].

UKF parameters		
α	β	κ
1.3	1.5	0.2



(a) k=0 (b) k=10

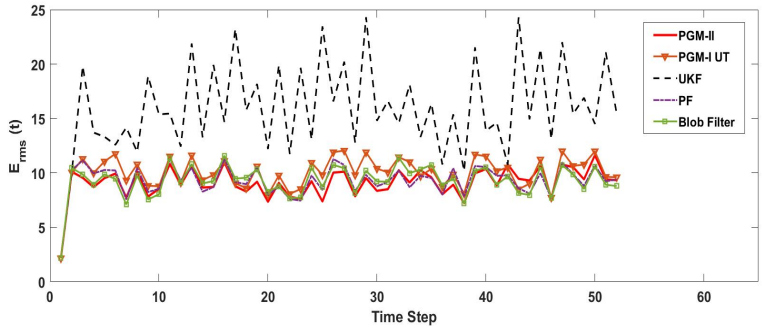
(c) k=25 (d) k=30

(e) k=40 (f) k=50

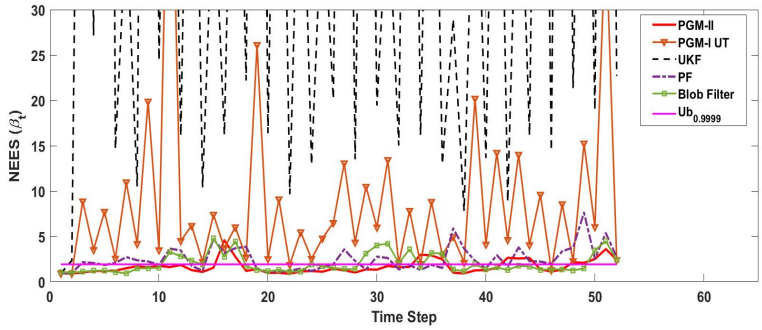
Figure 5.2: Evolution of state PDF with time

employed with 80 particles and a maximum number of 6 mixture components. For the blob filter, 80 Gaussians with a maximum covariance of of 10^{-4} was used in the estimation process.

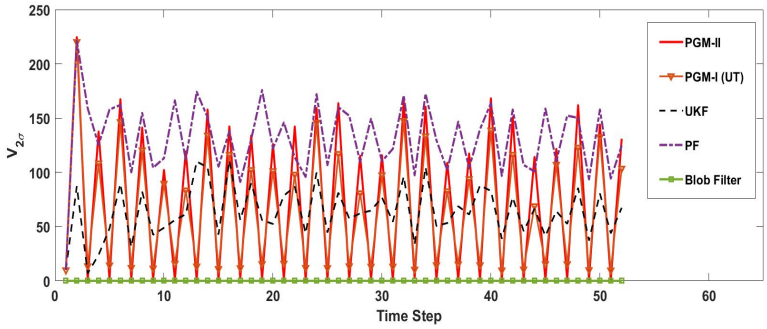
The Monte Carlo averaged RMSE results (E_{rms}) are plotted in Fig. 5.3a. The PGM-II filter, blob filter and the PF are seen to outperform the UKF by a large margin. The tracking performance of PGM-II filter is also found to be somewhat



(a) Monte Carlo averaged root mean squared error ($E_{rms}(t)$)



(b) Monte Carlo averaged NEES test statistic(β_t)



(c) Average volume of $2-\sigma$ ellipse ($V_{2\sigma}(t)$)

Figure 5.3: PGM-II estimation results for one dimensional multimodal system. Reprinted with permission from [3, 4].

better than that of PGM-I filter. The time averaged tracking error $\overline{E_{rms}}$ given in Table 4.2 underlines this observation. The results of NEES test plotted in Fig. 5.3b show that the UKF estimates overstep the 99.99% upper bound $Ub_{0.9999}$ during the entire duration of the simulation after $t = 1$. The PGM-II filter and the blob

filter are seen to offer more consistent estimates that lie within the 99.99% upper bound. The total fraction of the simulated time ($\beta_{c\%}$) during which each filter offered consistent estimates according to the $Ub_{0.9999}$ can be computed. The values of $\beta_{c\%}$ for all three filters are also listed in Table 4.2. The results indicate that the PGM-II filter outperforms the blob filter, PF, PGM-I filter and the UKF. Finally, the Monte Carlo averaged $2-\sigma$ volumes for each of the three filters are plotted in Figure 5.3c. The time averaged values of the $2-\sigma$ volumes are listed in Table 5.2. The blob filter is seen to have the smallest time averaged $2-\sigma$ volumes.

5.5.2 Example 2

In this example, we evaluate the performance of PGM-II filter in the so called "Blind tricyclist" problem proposed in [15]. As the name suggests, the Blind tricyclist problem involves the estimation of the state of a blind tricyclist steering across an amusement park. The blind tricyclist is given the speed and steering angle time histories as inputs so that he can navigate across the park. However, his initial position coordinates (X_1, X_2) and heading angle (X_3) are unknown to him. To assist the navigation, measurements are recorded, but only intermittently and they consist of the relative bearing angle between the tricyclists heading and the location of two friends who are riding merry-go-rounds. The blind tricyclist can distinguish between

Table 5.2: Time averaged RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for PGM-II, test case 1. Reprinted with permission from [3, 4].

Case 1 Results			
	$RMSE_{pos}$	%cases above $99.99\%U_b$	$V_{2\sigma}$
PGM-II	9.1066	25	71.8883
PGM-I(UT)	10.0047	92.31	64.3722
UKF	15.9752	98.08	63.4955
PF	9.2925	59.62	131.5503
Blob Filter	9.2737	34.62	0.0137

the measurements coming from the two friends. However he only knows the centers and radii of the merry-go-rounds with certainty. The initial rotation angles (X_4, X_5) and the fixed rotation rates (X_6, X_7) of the two merry-go-rounds are unknown. The objective of the blind tricyclist problem is to estimate the quantities $X_1, \cdots, X_7$ at all times. Hence it is a seven dimensional nonlinear estimation problem that involves both static and dynamic parameters. The equations governing the evolution of the state variables can be found in [15]. Figure 5.4 shows a realization of the truth trajectory of the blind tricyclist on the ground along with two merry-go-rounds.

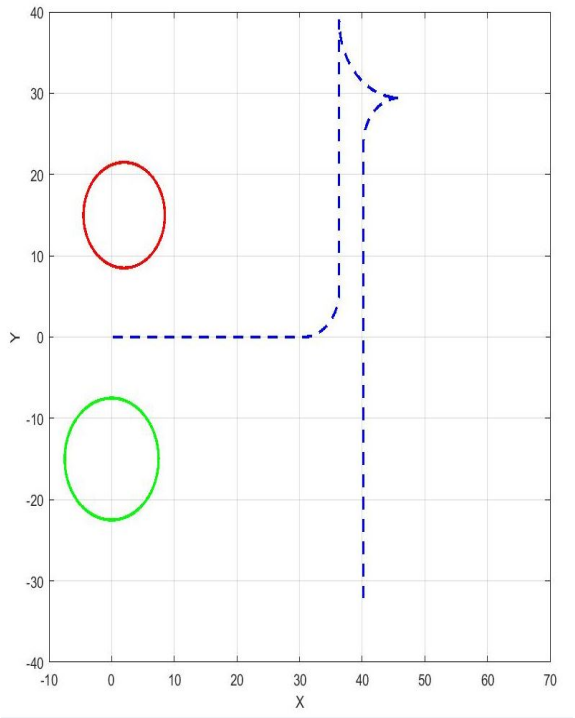


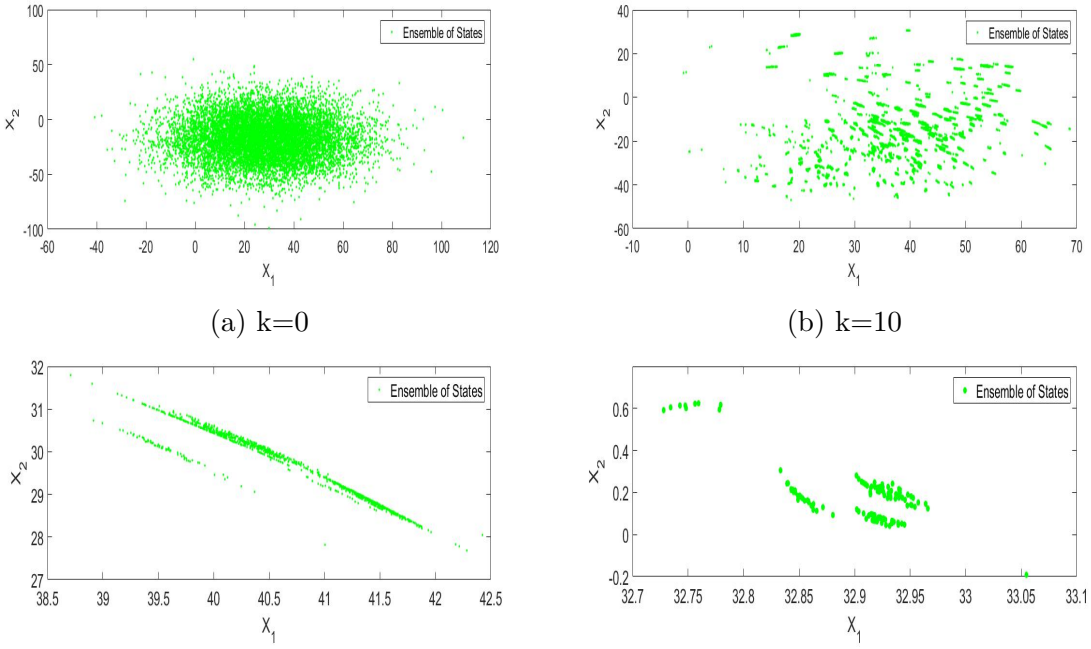
Figure 5.4: Zero process noise ground trajectory of the blind tricyclist and the location of merry-go-rounds

The relative bearing angle measurement between the blind tricyclist and the first

merry-go-round at the instant k is given by

$$\begin{aligned}\psi_{1,k} =& atan2(\{(y_1 + \rho_1 \sin(X_4) - X_2 - b_r \sin(X_3)), \\ & (x_1 + \rho_1 \cos(X_4) - X_1 - b_r \cos(X_3))\} - X_3 + \nu_k.\end{aligned} \tag{5.47}$$

Here, (x_1, y_1) represents the center of the first merry-go-round, ρ_1 represents its radius and b_r represents the distance between the point below the blind tricyclists head and the midpoint of the two rear wheels. From this, it is clear that multiple sets of the quadruple (X_1, X_2, X_3, X_4) can result in same value for the measurement $\psi_{1,k}$ even in the absence of the Gaussian noise ν_k. The state PDF in the blind tricyclist problem is highly multimodal due to this measurement ambiguity. Figure 5.5 shows the marginal distribution of the position coordinates (X_1, X_2) at different time steps, evaluated using a blob filter simulation. The initial uncertainty is assumed to be a unimodal Gaussian as seen in Figure 5.5a. By the end of the tenth time step, the unimodal state PDF is seen to have split into several modes distributed over a large area. As time progresses, the mixture weights also evolve and only a few of these modes survive. Plotted in Figure 5.5c is the distribution of states at the time step $k = 120$. The states are seen to be distributed between two major clusters. However at $k = 230$, we see four different clusters. Thus not only is the blind tricyclist problem multimodal, but the number of modes is seen to vary widely between different time steps. Additionally, new modes are seen to be created during the measurement update step. This affects the applicability of filters with Kalman type correction as the number of components remain intact during the update even in mixture Kalman implementations. The noise parameters used in simulating the blind tricylist problem are given in [15].



(a) k=0 (b) k=10

(c) k=120 (d) k=230

Figure 5.5: Marginal distribution of X_1, X_2

Table 5.3: UKF Parameter Values for PGM-II, test case 2. Reprinted with permission from [3, 4].

UKF parameters		
α	β	κ
0.01	2	0

The PGM-II filter, the UKF, the PF and the blob filter are employed in the estimation of the blind tricyclist problem. The PGM-I and PGM-II filters are implemented with 8000 particles where as 10000 particles were used in the SIR type PF implementation. The values of the parameters α, β, χ used in the UKF implementation are listed in Table 5.3. The blob filter was implemented using 7000 Gaussian PDFs with the LMI upper bound on the mixture covariances chosen from [22]. For

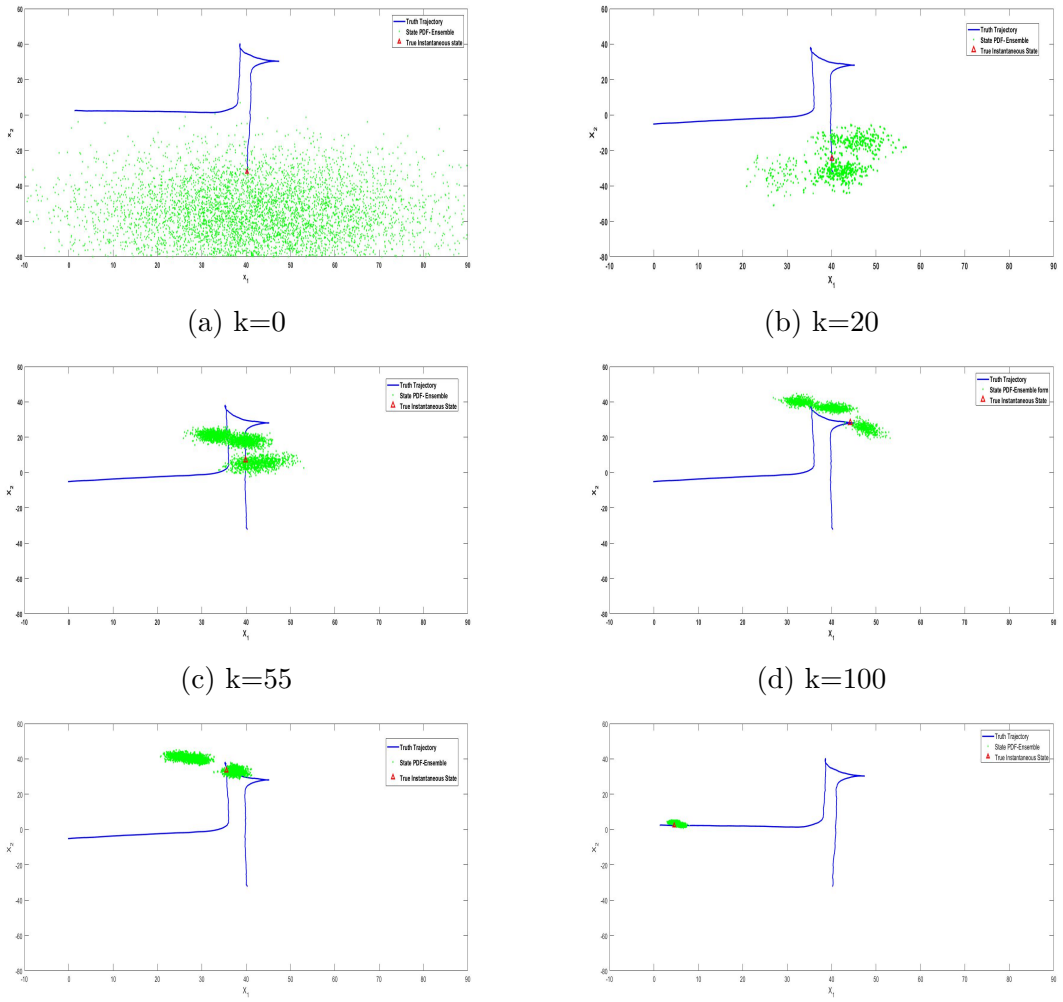
the MCMC step, the length of the burn-in time is set to be 800. The sampling covariance was chosen as $0.05 \times P_i^-(n)$ where $P_i^-(n)$ represents the i^{th} propagated prior covariance given by the clustering algorithm. The maximum number of mixture components used during the clustering step is chosen to be three. However, in order for the filter to not assign disproportionate confidence in any single mode, the diagonal elements of the clustered prior covariance matrices are never allowed to fall below a certain lower bound. This helps to prevent the loss of diversity. It also makes the estimates less accurate. When the diagonal elements do fall below this threshold, they are updated artificially. The lower bounds used on the diagonal elements of the prior covariance are summed up in the vector V_{lb} below.

$$V_{lb} = \begin{bmatrix} 280 & 280 & 0.4 & 9.9 & 9.9 & 0.02 & 0.02 \end{bmatrix}^T \times 10^{-2} \tag{5.48}$$

The marginal PDF in X_1, X_2 estimated using PGM-II filter at different time steps is plotted in Figure 5.6. The true state X_1, X_2 is also shown for reference. The mixture modes are seen to undergo splitting and merging during the course of estimation.

The accuracy and informativeness of the estimation results are analyzed using RMSE and $V_{\sigma 2}$ as in test case 1. However, the NEES test is performed as described in [15], i.e, by computing the fraction of the total number of Monte Carlo runs that produced NEES test statistic that falls within the 99.99% upper bound of a seven dimensional chi squared random variable. This upper bound is computed to be equal to $U_b = 29.8775$. The RMSE results obtained from 50 Monte Carlo runs of the blind tricyclist problem are plotted in Figure 5.7.

The results show that by the end of the estimation process, the blob filter offers the most accurate estimates followed by the PGM-II filter. The terminal RMSE



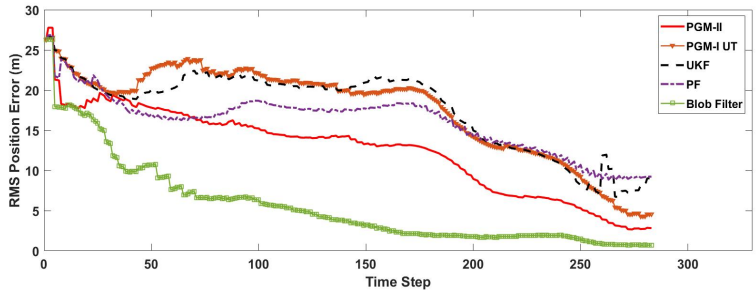
(a) k=0 (b) k=20

(c) k=55 (d) k=100

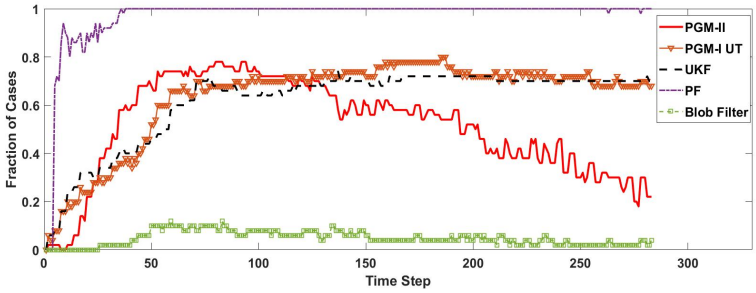
(e) k=180 (f) k=270

Figure 5.6: Evolution of the marginal distribution of X_1, X_2 estimated using PGM-II filter

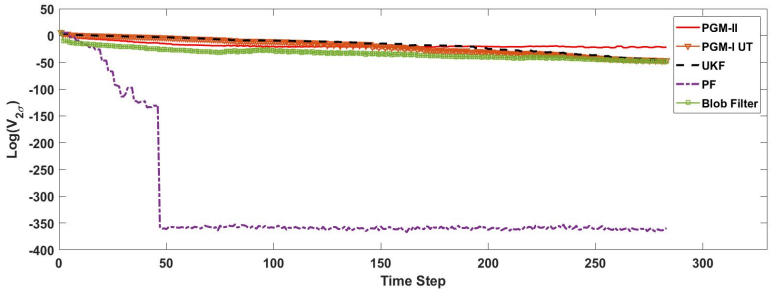
position error, terminal % of cases where the NEES results are above 99.99% and the time averaged 2 sigma ellipse volume $V_{\sigma 2}$ are provided in Table 5.4. The NEES result plotted in Figure 5.7b show that the blob filter offers the most consistent results among the three filters. The terminal NEES results presented in Table 5.4 show that the blob filter provided consistent terminal estimates in 96% of the simualtions. This was followed by the PGM-II filter and the UKF. The UKF is seen to provide more



(a) Root mean squared position error



(b) NEES Results



(c) Logarithm of the Monte Carlo averaged 2σ ellipse volumes

Figure 5.7: PGM-II estimation results for Blind tricyclist problem. Reprinted with permission from [3, 4].

consistent estimates than the PGM-II during some window of the simulated time. However it must be observed that the UKF covariances are seen to be nearly 4 orders of magnitude larger than the PGM-II on average. The value of $V_{\sigma 2}$ for the PF is seen to be smallest However, this result must be analyzed in conjunction with the fact that the PF results are almost always inconsistent.

The blob filter is seen to outperform the PGM-II filter in the estimation of the

Table 5.4: Time averaged RMSE $\overline{E_{rms}}$, likelihood $\hat{L}$, and the 2σ volume for PGM-II, test case 2. Reprinted with permission from [3, 4].

Case 2 :Terminal Results			
	$RMSE_{pos}$	%cases above 99.99%U_b	$\hat{log}(V_{2\sigma})$
PGM-II	2.8257	22	-21.8973
PGM-I(UT)	4.5577	68	-46.4367
UKF	9.0014	70	-45.2474
PF	9.2239	100	-359.4890
Blob Filter	0.6999	4	-48.2278

blind tricyclist problem. This indicates certain important limitations associated with the implementation of PGM-II filter. In theory, the MCMC based measurement update is capable of sampling from any posterior probability distribution. It is also well suited for sampling in large dimensions in comparison to other approaches such as the importance sampling. However, when the target distribution is extremely multimodal as in the blind tricyclist problem, the Markov chain is prone to being trapped in one of the modes. This can diminish the ability of the MCMC based approaches to sufficiently explore the state space in a reasonable amount of time. The parallelized approach presented in this work was meant to alleviate this problem. The results indicate that this aspect of the problem requires further study. Increasing the number of clusters is one possible solution. However, when implemented sequentially, each new cluster results in an increased cost of at least one full burn-in time. Furthermore, the mixture models obtained from the samples may not accurately represent the component densities as the perfect clustering assumption may not hold in practice. It must be observed that while the LMI based 'blob' filtering approach has several advantages over the conventional Gaussian sum filters [19], the number of Gaussians used may still need to be increased exponentially with the dimension of the state space in order to cover the volume of a single Gaussian during re-approximation.

Table 5.5: Computation time per filtering run.

Computation Time (s)			
UKF	PF	Blob Filter	PGM-II
0.2630	49.3090	64.0793	565.2952

We also looked at the computation time required by each filter to complete the estimation of the blind tricyclist problem. The computation time per filtering run as returned by the timeit() function from matlab on a 3.2 GHz PC running Windows is presented in Table 5.5. The PGM-II filter is seen to require the most computational resources. This is partly due to the large number of samples the MCMC based measurement update requires to explore the state space when the PDF is extremely multimodal.

5.6 Discussion

In this chapter, we developed a PGM-II filter for the general multimodal nonlinear filtering problem Our results demonstrate that the PGM-II filter is capable of handling the nonlinear/non-Gaussian measurement update. Having developed the PGM filters, we now intend to test their performance against other filters that have been proposed to address the general nonlinear filtering problem. The results of these experiments and the related discussion can be found in the next chapter.

6. TESTS AND EXTENSIONS

6.1 Benchmark Performance Tests

In this section, we extensively test and compare the estimation performance of PGM filters with other nonlinear filters. We consider a selection of problems that have been used to evaluate the performance of other general purpose filters in recent literature. We intend to investigate the relationship between the estimation performance of the PGM filters and dimensionality. We shall also consider the effect of the number of particles on PGM filtering performance.

6.1.1 Coupled Multi-Target Tracking Model

In this example, we consider the coupled multi-target tracking model problem described in [53]. This model describes the dynamics of a group of targets that execute coupled nonlinear motion on a 2D surface. The complete set of equations describing the system are given below (equations 6.1-6.14). The state vector of each target consists of its x, y position and corresponding velocity coordinates. As a result, in the n-target test case, the dimensionality (d) of the problem becomes 4n. Note that the dynamics of the targets are coupled through the terms $\Pi^i_{x_k}, \Pi^i_{y_k}$. The strength of this coupling depends on the parameters $\kappa_1, \kappa_2, \kappa_3$. When all three coupling parameters are set to zero, the system will reduce to n independently moving targets. Hence the effective dimensionality of the system depends highly on the values

of $\kappa_1, \kappa_2, \kappa_3$.

$$x^i_{k+1} = x^i_k + \dot{x}^i_k \Delta t + \frac{1}{2} a_{x_{k+1}} \Delta t^2 \tag{6.1}$$

$$y^i_{k+1} = y^i_k + \dot{y}^i_k \Delta t + \frac{1}{2} a_{y_{k+1}} \Delta t^2 \tag{6.2}$$

$$\dot{x}^i_{k+1} = \dot{x}^i_k + \Pi^i_{x_k} \Delta t + a_{x_{k+1}} \Delta t \tag{6.3}$$

$$\dot{y}^i_{k+1} = \dot{y}^i_k + \Pi^i_{y_k} \Delta t + a_{y_{k+1}} \Delta t \tag{6.4}$$

$$\Pi^1_{x_k} = \frac{1}{N-1} \sum_{i=2}^{N} \Big(\frac{\kappa_1}{\sqrt{(x^1_k - x^i_k)^2 + (y^1_k - y^i_k)^2} + \delta} \Big) \frac{v_t^2}{r_t} \cos\big(\frac{v_t}{r_t} k\big) \tag{6.5}$$

$$\Pi^1_{y_k} = \frac{1}{N-1} \sum_{i=2}^{N} \Big(\frac{\kappa_1}{\sqrt{(x^1_k - x^i_k)^2 + (y^1_k - y^i_k)^2} + \delta} \Big) \frac{v_t^2}{r_t} \sin\big(\frac{v_t}{r_t} k\big) \tag{6.6}$$

$$\Pi^i_{x_k} = \kappa_2 (x^1_k - x^i_k) - \kappa_3 \dot{x}^i_k \tag{6.7}$$

$$\Pi^i_{y_k} = \kappa_2 (y^1_k - y^i_k) - \kappa_3 \dot{y}^i_k \tag{6.8}$$

$$r^i_{k+1} = \sqrt{(x^1_k)^2 + (y^1_k)^2} + \nu^i_{r_{k+1}} \tag{6.9}$$

$$\theta^i_{k+1} = \arctan \frac{y^i_k}{x^i_{k+1}} + \nu^i_{\theta_{k+1}} \tag{6.10}$$

$$\begin{aligned} p(z_{k+1}|x_{k+1}) &= p(r_{k+1}|x_{k+1}) p(\theta_{k+1}|x_{k+1}) \\ &= \frac{1}{(2\pi\beta^2)^{\frac{N}{2}} |R_r|^{\frac{1}{2}}} \exp\big\{ -\frac{1}{2} (r_{k+1} - \tilde{r}_{k+1})^T R_r^{-1} (r_{k+1} - \tilde{r}_{k+1}) \big\} \end{aligned} \tag{6.11}$$

$$\times \prod_{i=1}^{N} exp\Big\{ -\frac{1}{\beta} \big(\theta^{(i)}_{k+1} - \arctan\big(\frac{y^{(i)}_{k+1}}{x^{(i)}_{k+1}}\big) \big) \Big\} \tag{6.12}$$

$$\tilde{r}_{k+1} = \begin{bmatrix} \sqrt{(x^1_k)^2 + (y^1_k)^2} & \sqrt{(x^2_{k+1})^2 + (y^2_{k+1})^2} & \cdots & \sqrt{(x^N_{k+1})^2 + (y^N_{k+1})^2} \end{bmatrix} \tag{6.13}$$

$$R_r = \begin{bmatrix} \sigma_r^2 & \sigma_{r_x}^2 & \cdots & \sigma_{r_x}^2 \\ \sigma_{r_x}^2 & \sigma_r^2 & \cdots & \sigma_{r_x}^2 \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{r_x}^2 & \sigma_{r_x}^2 & \cdots & \sigma_r^2 \end{bmatrix} \tag{6.14}$$

Estimation of the coupled multi-target system is to be performed using the range

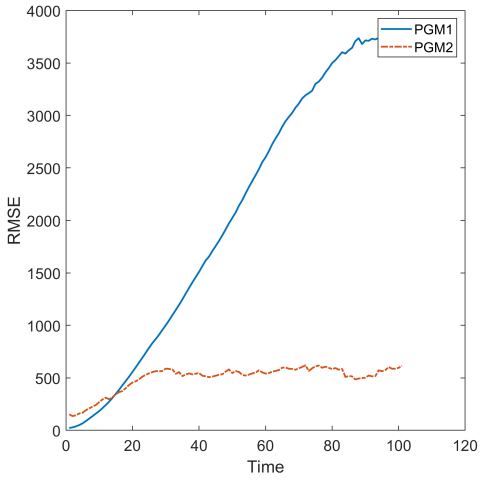
and bearing measurement model described in equations 6.9 and 6.10. The process model and the range measurements are assumed to be perturbed by Gaussian noise terms. A positive correlation is assumed to exist among range noise terms. However, for bearing measurements we assume a non-Gaussian exponential measurement noise as given in equation 6.11. Note that, in order to perform the PGM-1 update, one needs the distribution parameters such as μ_Y, P_{YY} etc of the measurement random variable. So incorporating non-zero mean, non-Gaussian measurement noise terms require a modification of the standard PGM-1 Kalman update. The standard PGM-1 update uses noiseless measurements to compute the mean μ_Y and covariance P_{YY} of measurements. The measurement noise covariance R is then added afterwards to P_{YY}. In this example, we used noise perturbed measurements directly to compute the statistics of the measurements.

In [53], the authors evaluate the performance of a new log homotopy based particle flow filter on the coupled multi-target tracking model in terms of a non-dimensionalized RMSE metric ϵ_d. Let M be the total number of simulations, n be the total number of targets. Let $(x_k^{i,m}, y_k^{i,m}), (\hat{x}_k^{i,m}, \hat{y}_k^{i,m})$ be the estimated and true position coordinates of the i^{th} target at k^{th} time step during m^{th} Monte Carlo run. Then the non-dimensionalized RMSE metric ϵ_d is given by

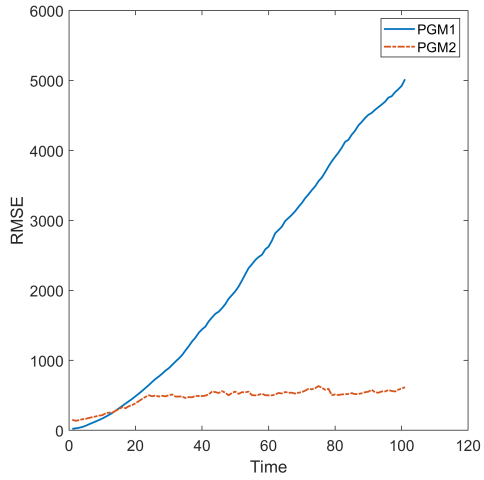
$$\epsilon_d = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\Big[\frac{2}{n}\sum_{i=1}^{n}\Big(\big(x_k^{i,m}-\hat{x}_k^{i,m}\big)^2+\big(y_k^{i,m}-\hat{y}_k^{i,m}\big)^2\Big)\Big]}. \tag{6.15}$$

In this experiment, we evaluate the performance of the PGM filters in terms of ϵ_d as the number of targets and the level coupling is adjusted.

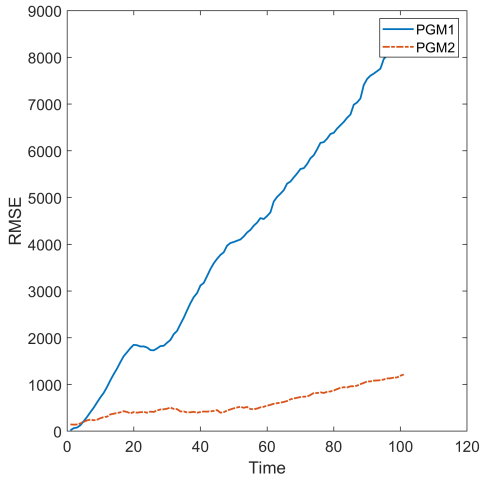
Coupling : From the equations, it can be seen that the coupling between the dynamics of different targets depends on parameters $\kappa 1, \kappa 2, \kappa 3$.



(a) Uncoupled Model ($\kappa 1 = 0$, $\kappa 2 = 0$, $\kappa 3 = 0$)



(b) Weakly coupled Model ($\kappa 1 = 100$, $\kappa 2 = 0.005$, $\kappa 3 = 0.005$)



(c) Strongly coupled Model($\kappa 1 = 8000$, $\kappa 2 = 0.05$, $\kappa 3 = 0.1$)

Figure 6.1: Comparison of the estimation performances of PGM-I and PGM-II filters for uncoupled and coupled models.

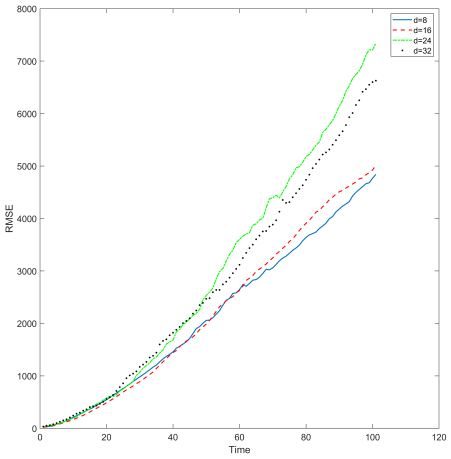
We investigate the estimation performance of the PGM filters in three different scenarios corresponding to three different combinations of coupling parameters. These are (1) Uncoupled model (Figure 6.1a) (2) Weakly coupled model (Figure 6.1b)

and (3) Strongly coupled model (Figure 6.1c) [53]. We set the number of targets to four and run the PGM filters to track the targets for 100 steps. We evaluate the PGM estimation errors for each of the three different coupling scenarios. In each case, the experiment is repeated 50 times and the average error is plotted for comparison. We use 6000 particles in our simulations.

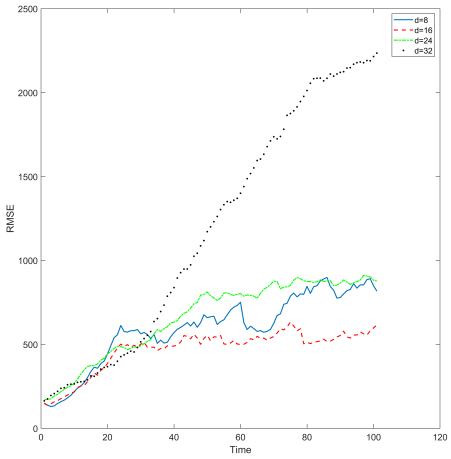
The results plotted in Figures 6.1a, 6.1b, 6.1c show that, for a given number of particles, the estimation performance deteriorates when the coupling is increased. A similar effect is observed in all the filters that are studied in [53]. This is expected as the targets can be estimated separately and independently when they are uncoupled. In that scenario, we can track the states n targets at a given level of accuracy while using only as many number of points as necessary to track the state of a single target. With coupling however, the relationship between different components of the state vector becomes stronger as the dynamics of the targets become more intertwined. This results in an increased computational demand to achieve a given level of estimation accuracy. Among the several log homotopy based particle flow filters that are considered in [53], a non-zero diffusion constrained flow-Daum Huang filter(NZDCF-DHF) is found to offer the best tracking performance. In the uncoupled scenario, both PGM-1 and PGM-II filters outperform the NZCDF-DHF filter with the PGM-II filter outperforming it by a considerable margin. The PGM filters are both found to perform better than a bootstrap particle filter that uses 2.5×10^5 particles as well. The performance of PGM-1 and NZCDH-DHF are more matched in the weakly coupled scenario. PGM-1 filter is also outperformed by the boot strap filter in this case. However, PGM-II filter still maintains a considerable margin in its performance in comparison to other filters. As the coupling is made strong, NZCDH-DHF performance is found to become better than that of PGM-1 Filter. However, PGM-II filter remains considerably more accurate than all other filters.

The terminal error of PGM-II filter in the strongly coupled scenario is found to be near 1000 where as it is found to fall between 4000 and 8000 for the NZCHF- CDF. Note that log homotopy filters use the exact Bayesian update equations to derive the differential equations governing the particle flow. This may be the reason for the slower deterioration of their performance in comparison to PGM-1 filter as the system becomes more coupled and nonlinear. Overall, the PGM-II filter is found to outperform every other filter in all cases considered.

Number of Targets : In this experiment, we study the effect of increasing the number of targets on PGM filtering performance. The PGM filters are implemented with 6000 particles and the tests are conducted starting with a minimum of two targets ($d = 8$). The estimation was performed over 100 time steps and repeated over 50 Monte Carlo runs. The results are plotted in Figure 6.2a, 6.2b. For the PGM1 filter, increasing the dimensions from 8 to 16 did not produce a major increase in RMSE. However, raising the dimensions further to 24 resulted in an appreciable difference in estimation error. For PGM-II filter, the variation in error was found to be minor from $d = 8$ to $d = 24$. However, increasing the dimensions further to 32 resulted in significant increase in estimation error. Clearly both filters demonstrate a clear dimension dependence when it comes to estimation performance. The more accurate measurement update step in PGM-II filter allows it to use a smaller number of particles and achieve more accurate results than PGM-1. The large jump in PGM-II error may be due to the MCMC requiring a much larger number of samples to explore the 32 dimensional state space.



(a) PGM-I filter



(b) PGM-I filter

Figure 6.2: Growth in Monte Carlo averaged estimation error with increasing dimensions for PGM-I and PGM-II filters.

6.1.2 Generalized Bimodal System Model

In this case, we consider a generalization of the standard one dimensional bimodal estimation problem [23]. The system considered herein, obtained by coupling several one dimensional bimodal nonlinear systems, is found in [54]. The unperturbed system dynamics is given by

$$\begin{aligned} x_{k+1} =& 0.5x_k + 8\cos 1.2(k-1) \\ &+ \begin{cases} 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^{c})^2}, & if c = 1 \\ 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^{c-1})^2}, & if 1 < c < d \\ 2.5\frac{x_{k-1}^{c}}{1+(x_{k-1}^{c-1})^2}, & if c = d \end{cases} \end{aligned} \tag{6.16}$$

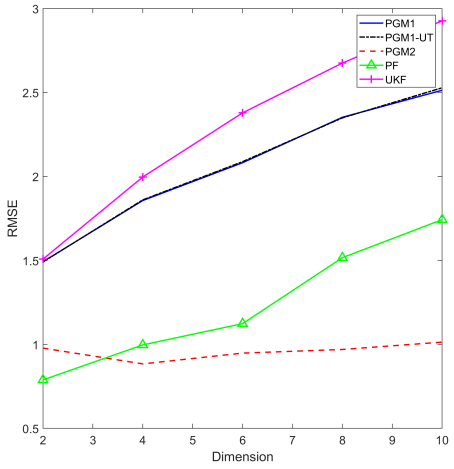
Note that this model allows us to freely choose d. An additive multi-modal process noise $v_k \sim \sum_{i=1}^{3} \frac{1}{3}\mathcal{N}(\mu_i \mathbb{1}_{d\times 1}, \sigma_v^2 \mathbb{I}_{d\times d})$ with $\mu_1 = -1$, $\mu_2 = 0$,$\mu_3 = 1$ and $\sigma_v = 0.5$ is

assumed. Measurements are recorded as

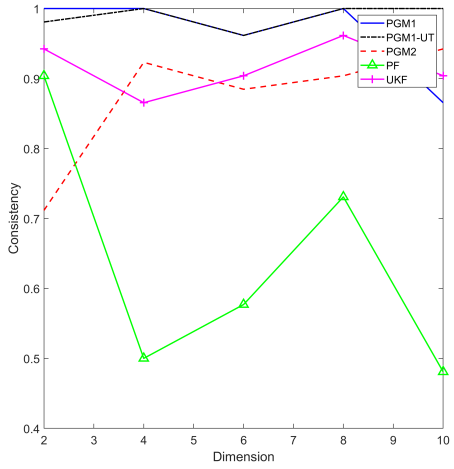
$$y_k = \frac{(x_k^c)^2}{20} + \omega_k \tag{6.17}$$

with $\omega_k \sim \sum_{i=1}^{3} \frac{1}{3}\mathcal{N}(\delta_i \mathbb{1}_{d\times 1}, \sigma_\omega^2 \mathbb{I}_{d\times d})$ with $\delta_1 = -3$, $\delta_2 = 0$, $\delta_3 = 3$ and $\sigma_\omega = 0.1$. Note that both the unperturbed process/measurement models and the random noise terms contribute to the multimodality in this problem. We study the changes in the estimation performance of the PGM filters as the number of dimensions is increased from $d = 2$ to $d = 10$. The estimation performance is evaluated in terms of RMSE, the NEES test metric, Avg.likelihood and Avg.2-σ volume of state PDF. Once again, we have a system that has non-Gaussian measurement noise. An M component Gaussian mixture measurement noise model will split the product term in the Bayesian update into M separate terms. In the context of PGM filtering, this will result in a straightforward generalization of PGM-1 filters that involves M separate Kalman measurement updates. A non-Gaussian measurement noise can also be incorporated by sampling the measurement noise terms as we did in the previous example. Doing this is equivalent to obtaining a unimodal Gaussian approximation of the measurement noise random variable via matching of the first two moments. In order to test the performance of our existing PGM-1 filter design, we shall approximate the 3 component measurement noise model with a single Gaussian noise model through the moment matching approach. We compare the performance of the PGM filters with that the of the UKF and PF. The estimation results obtained from this experiment are given in Appendix B.1. At $d = 2$, the PF is seen to perform slightly better than PGM-II in terms of accuracy as at 8000, the number samples used is sufficiently high for a 2-dimensional estimation problem. However, as the dimensions increase, both the accuracy and consistency of PF estimates is observed

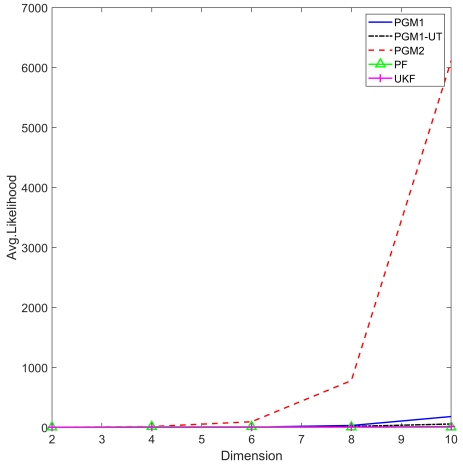
to decline while the PGM-II filter is seen to offer the most accurate results. It also dominates in terms of average likelihoods. Interestingly, filters that use the Kalman measurement update are seen to offer more consistent results than those that rely on sampling. In particular,the PGM-1 filters are seen to offer the most consistent estimates in all experiments. Additionally, PGM-II results are found to be signifi-



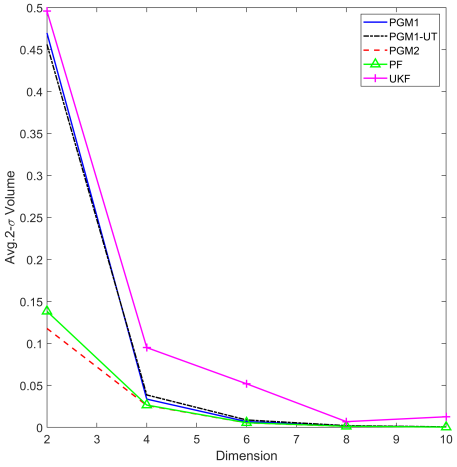
(a) RMSE



(b) NEES



(c) Likelihood



(d) $2-\sigma$ Volume

Figure 6.3: Time averaged performance metrics for the bi-modal system model.

cantly more consistent than that of the PF. These trends become clearer in the time averaged performance metrics that are plotted in Figure 6.3.

6.1.3 Re-entry Problem

In this experiment, we estimate the state of a vehicle that enters the earth's atmosphere from space. This five dimensional problem was previously used to test the performance of feedback particle filter (FPF) [55] and UKF [13]. Like the blind tricyclist problem, it involves the estimation of both dynamic ($x_1 - x_4$) and static (x_5) parameters. The vehicle is assumed to be subject to gravity and drag force in addition to random perturbations. The equations of motion for this system are given below

$$\dot{x}_1 = x_3 \tag{6.18}$$

$$\dot{x}_2 = x_4 \tag{6.19}$$

$$\dot{x}_3 = Dx_3 + Gx_1 + \omega_1 \tag{6.20}$$

$$\dot{x}_4 = Dx_4 + Gx_2 + \omega_2 \tag{6.21}$$

$$\dot{x}_5 = \omega_3 \tag{6.22}$$

$$D = bexp(\frac{r_0 - r}{h_0})\nu \tag{6.23}$$

$$b = b_0 exp(x_5) \tag{6.24}$$

$$r = \sqrt{x_1^2 + x_2^2} \tag{6.25}$$

$$v = \sqrt{x_3^2 + x_4^2} \tag{6.26}$$

$$G = -\frac{\mu}{r^3} \tag{6.27}$$

$$r_0 = 6374, b_0 = -0.59783 \tag{6.28}$$

$$h_0 = 13.406, \mu = 398601.2 \tag{6.29}$$

The random perturbations $\{w_i\}_{i=1}^{3}$ are modeled as Gaussian white noise processes with zero mean and power spectral density

$$Q = \begin{bmatrix} 2.4064 \times 10^{-4} & 0 & 0 \\ 0 & 2.4064 \times 10^{-4} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.30}$$

Note that the static parameter is perturbed by the noise term ω_5. In order to enhance the estimation of the static parameter, [55] recommends setting a variance of 10^{-5} for w_3. A range and bearing measurement model is assumed,

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan \frac{x_2}{x_1} \end{bmatrix} + \nu \tag{6.31}$$

The system is simulated for a duration of 200 s with the integration time step set at $dt = 0.01s$. The initial state distribution is assumed to be Gaussian with mean

$$\mu_0 = \begin{bmatrix} 6500.4 & 349.14 & -1.8093 & -6.7967 & 0.6932 \end{bmatrix} \tag{6.32}$$

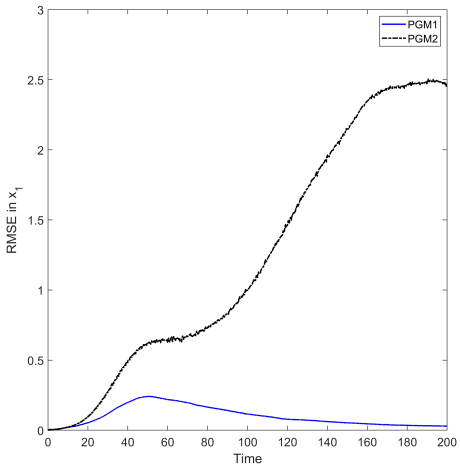
and covariance

$$P_0 = diag(\begin{bmatrix} 10^{-6} & 10^{-6} & 10^{-6} & 10^{-6} & 0 \end{bmatrix}. \tag{6.33}$$

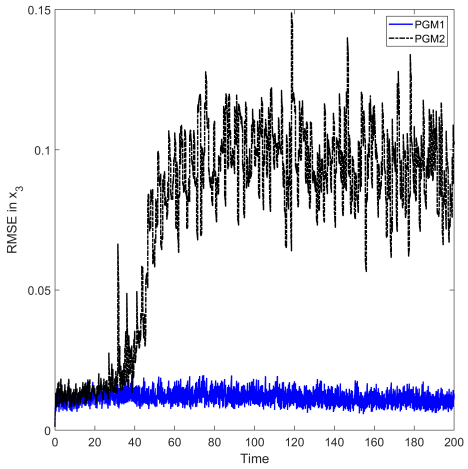
Measurements are recorded at the interval of 50 time steps. The measurement noise term is assumed to be Gaussian distributed with zero mean and covariance

$$R = diag(\begin{bmatrix} 10^{-6} & (\frac{0.01\pi}{180})^2 \end{bmatrix}). \tag{6.34}$$

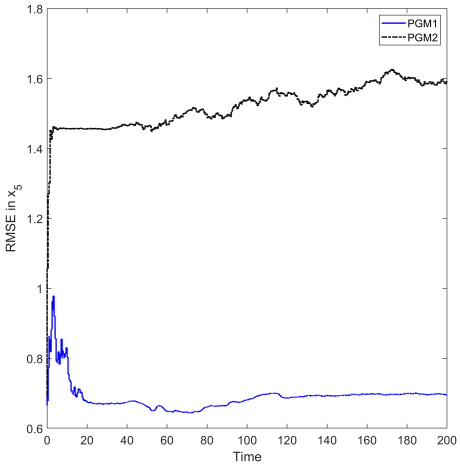
In order to investigate the relationship between estimation performance and the



(a) x_1



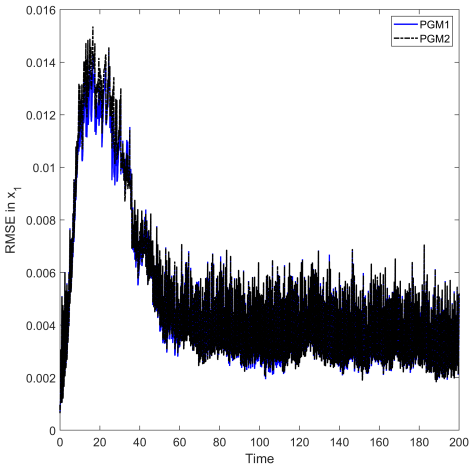
(b) x_3



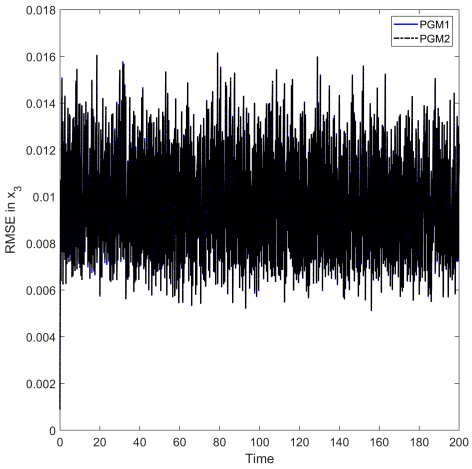
(c) x_5

Figure 6.4: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 10$

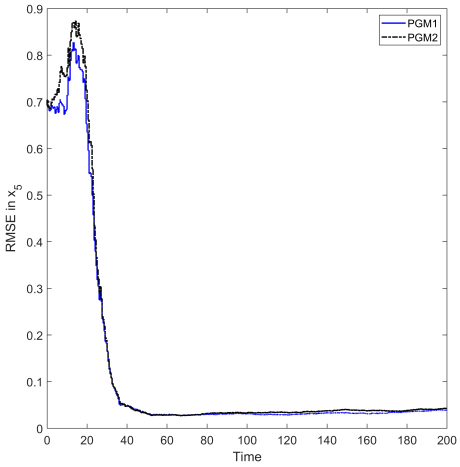
number of particles used (N_p), we conduct the experiment at N_p = 10,50,100,500,1000 and 5000. For each value of N_p, the experiment is repeated 50 times and the Monte Carlo averaged RMSE is computed. The Monte Carlo averaged RMSE for the states x_1, x_2 and x_5, for the cases when $N_p = 10$ and $N_p = 5000$ are plotted in Figures 6.4 and 6.5. The results for N_p = 50, 100, 500, 1000 are provided in Appendix B.2.



(a) x_1



(b) x_3

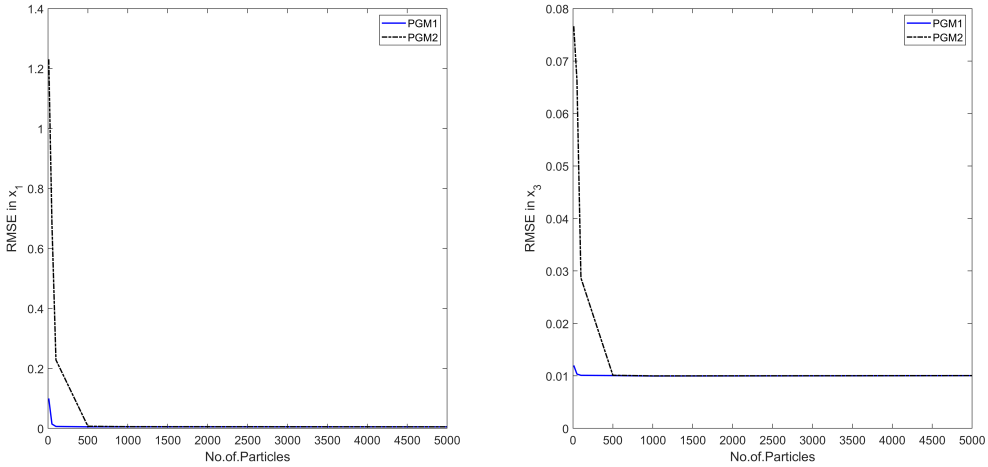


(c) x_5

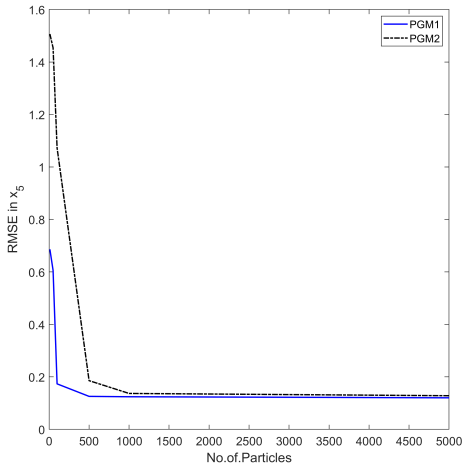
Figure 6.5: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 5000$

The results indicate that when the sample size is small, the PGM-1 filter outperforms the PGM-II filter when it comes to estimation accuracy in all three state variables. However, the performance of the latter improves substantially as the number of particles is increased. At around $N_p = 1000$, PGM-II can be observed to catch up with PGM-1 Filter as the RMSE plots for the former start to coincide with the

latter. The improvement in performance is summarized in Figure 6.6, in which we have plotted the RMSE against the number of particles, after performing a further averaging over time.



(a) x_1 (b) x_2



(c) x_3

Figure 6.6: Time Averaged RMSE in x_1, x_3 and x_5 for the Re-entry problem

We notice a precipitous drop in the averaged RMSE for PGM-II filter around

$N_p = 500$ and for PGM-1 filter for $N_p = 100$. The averaged error is seen to stabilize after $N_p = 500$ as increasing the numbers further up to 5000 does not produce substantial reduction in estimation error. The time averaged RMSE for both filters at $N_p = 10$ and $N_p = 5000$ are listed in Table 6.1.

Table 6.1: Time averaged RMSE for PGM-I and PGM-II for the re-entry problem

	Np=10			Np=5000		
	$\|\Delta x_1\|$	$\|\Delta x_3\|$	$\|\Delta x_5\|$	$\|\Delta x_1\|$	$\|\Delta x_3\|$	$\|\Delta x_5\|$
PGM1	0.0998	0.0120	0.6862	0.0050	0.0101	0.1202
PGM-II	1.2307	0.0766	1.5067	0.0051	0.0101	0.1282

Comparing the time averaged results for the PGM filters with the filters in [55], we find that PGM-1 filter outperforms the FPF at $N_p = 10$. However, the FPF is seen to perform better than PGM-II filter at N_p =10. This is expected as the MCMC method requires a much larger number of samples to capture the statistics of the state of this five dimensional system. However, as the number of samples is increased to $N_p = 5000$ the performance of the PGM-II filter is seen to be at par with that of PGM-1 filter. The performance of the FPF is also found to be similar to that of the PGM filters at $N_p = 5000$.

The results obtained from this section and from previous chapters indicate that despite their relative simplicity, the PGM filters can offer estimation performance that is at par with some of the latest such as the Blob filter, log homotopy based particle flow filters and the feedback particle filter. The performance of the PGM filters is found to be affected by the dimensionality and nonlinearity of the process and

measurement models. The Kalman measurement update used in PGM-1 filter is seen to offer highly consistent estimates even in large dimensions. Our results indicate that when the measurement model is highly nonlinear or when the measurement noise is non-Gaussian, PGM-II filter will offer superior estimation performance in comparison to the PGM-1 filter. However a generalization of the PGM-I filter that can handle Gaussian mixture measurement noise models can be obtained to improve the performance when measurement noise is multimodal. Additionally, when the computational resources are severely constrained and the number of samples that can be used are limited, it may be advisable to choose the PGM-I filter over PGM-II.

6.2 Augmented PGM-I Filter

In chapter 5, we proposed the PGM-II filter to handle the nonlinearity of the measurement function and non-Gaussianity of measurement likelihood in the filtering problem. The MCMC method used in the PGM-II update can, in theory, obtain samples from any probability density function. However, in practice, performing an MCMC update is significantly more expensive than the Kalman measurement update from a computational standpoint. In this section, we propose a modification of the PGM-I filter, namely the augmented PGM-I filter, that can better handle measurement nonlinearity and non-Gaussianity albeit being cheaper than the PGM-II filter.

First we shall describe the motivation behind the development of the augmented filter. We know that the exact Bayesian measurement update equation for computing the posterior PDF of state variable X_k conditioned on measurement $Z_k = z$ is

$$P(X_k|z) = \frac{P(X_k, Z_k = z)}{P(z)}. \tag{6.35}$$

The above relationship indicates that, given a functional representation of the joint PDF $P(X_k, Z_k)$, the posterior PDF can be computed by substituting the recorded measurement z into $P(X_k, Z_k)$ followed by normalization. However, in general, we may not have access to a functional/parametric representation of $P(X_k, Z_k)$. But, given the prior PDF $P(X_k)$ and the measurement model

$$Z_k = h(X_k) + \nu_k, \tag{6.36}$$

one can obtain samples from the joint PDF $P(X_k, Z_k)$ as follows

- Step 1: Sample $x_1 ... x_n$ from state PDF P(x)
- Step 2: Sample measurements $z_1 ... z_n$ by substituting state samples in the measurement function $h(x)$ and adding sampled measurement noise terms ν_k

The ordered pairs (x_i, z_i) thus obtained are distributed according to the joint PDF. Given a large enough sample, we can obtain an approximate functional representation of $P(X_k, Z_k)$ using an appropriate density estimation algorithm.

Let

$$P(X_k, Z_k) = \sum_{1}^{n} \omega_i P_i(X_k, Z_k). \tag{6.37}$$

In particular, we shall assume that the component PDFs $P_i(X_k, Z_k)$ can be accurately represented by a Gaussian PDF resulting in

$$P(X_k, Z_k) = \sum_{1}^{n} \omega_i \mathcal{N}([X_k, Z_k]', \mu_i, \Sigma_i). \tag{6.38}$$

Since $P_i(X_k, Z_k)$ is multivariate Gaussian PDF, the component posteriors $P_i(X_k | Z_k = z)$ obtained after Bayesian update will remain Gaussian. Note that this property will hold regardless of the nonlinearity of measurement function $h(x)$ and non-Gaussianity

of measurement noise ν_k. The component mean μ_i and covariance Σ_i appearing in equation 6.38 can be expanded as

$$\mu_i = \begin{bmatrix} \mu_{i,x} & \mu_{i,z} \end{bmatrix}' \tag{6.39}$$

$$\Sigma_i = \begin{bmatrix} \Sigma_{i,xx} & \Sigma_{i,xz} \\ \Sigma_{i,zx} & \Sigma_{i,zz} \end{bmatrix}. \tag{6.40}$$

The posterior component mean and covariance can be then computed as

$$\mu_{i,x|z} = \mu_{i,x} + \Sigma_{i,xz}\Sigma_{i,zz}^{-1}(z - \mu_{i,z}), \tag{6.41}$$

$$\Sigma_{i,xx|z} = \Sigma_{i,xx} - \Sigma_{i,xz}\Sigma_{i,zz}^{-1}\Sigma_{i,zx}. \tag{6.42}$$

Indeed, the standard Kalman update equations are a special case of the equations 6.41, 6.42 when Z_k is a linear function of X_k and ν_k is Gaussian distributed. The full posterior PDF can be expressed in terms of the component posteriors as

$$P(X_k|z) = \frac{\sum_1^n \omega_i P_i(z) P_i(X_k|z)}{P(z)}. \tag{6.43}$$

The term $P_i(Z)$ in the numerator can be calculated directly by marginalizing the component prior $P_i(X_k, z)$ and the constant term in denominator can be obtained as the normalizing constant. This completes the augmented PGM-I update.

In developing the augmented filter, we only assume that the joint PDF can be represented as a Gaussian mixture model. As long as this assumption is satisfied, and a density estimation algorithm that can accurately recover the joint PDF $P(X_k, Z_k)$ is available, we will be able to use the augmented PGM-I filter to handle nonlinearity/non-Gaussianity associated with measurement update. Since we have

relaxed the standard PGM-I assumption on Kalman update, the performance of the new filter will be only a function of the performance of the density estimation algorithm used. From the experiments in chapter 6, we discovered that a generalization of the PGM-I filter that incorporates Gaussian mixture measurement noise model can be derived to incorporate non-Gaussian measurement noise. However, this development can be foregone since we can use the augmented PGM-I filter for non-Gaussian noise models as long as it is straightforward to sample from.

We demonstrate the effectiveness of the augmented filter with a simple example.

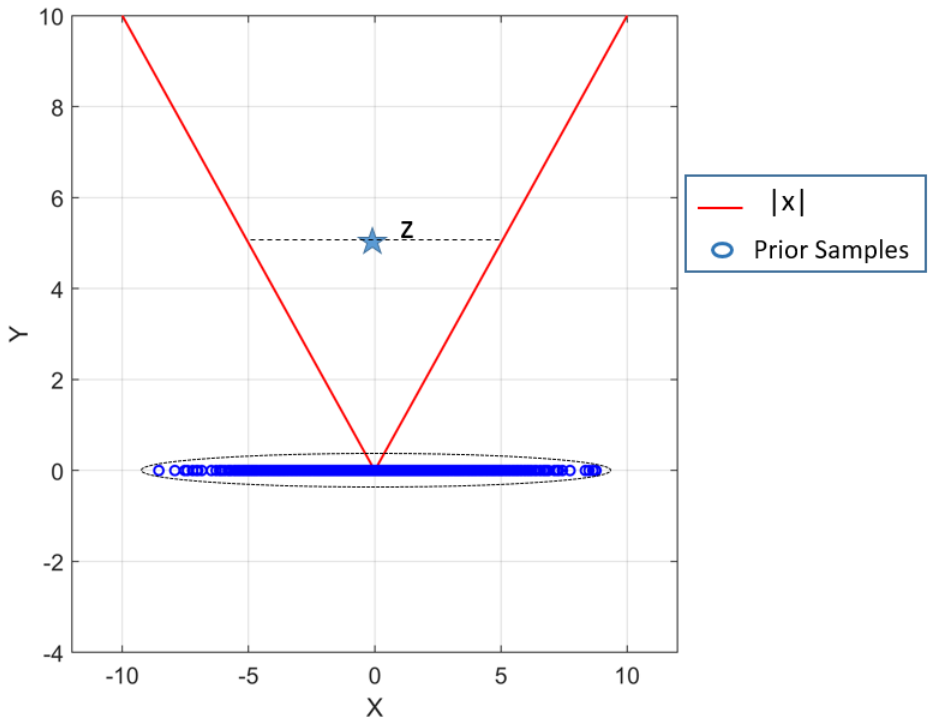


Figure 6.7: Optimal prior density estimate for PGM-I

Let $x \sim \mathcal{N}(0, 6)$. Let $h(x) = |x|$. We wish to find the posterior PDF of x given that a measurement $z = 5$ is recorded. The ideal density estimator for PGM-I filter, as

shown in Figure 6.7, will identify the unimodal Gaussian prior and attempt to perform Kalman measurement update over it, resulting in a unimodal posterior PDF. However since $|x| = 5$ has two roots, the true posterior will have two modes. On the other hand, the augmented filter, when used to perform measurement update will identify multiple components as shown in Figure 6.8. Not only does clustering help to identify the distribution of prior PDF, but given the augmented data, it can also help towards proper linearization of the measurement function by splitting the domain. Note that augmentation is also useful when the measurement noise is non-Gaussian and multimodal.

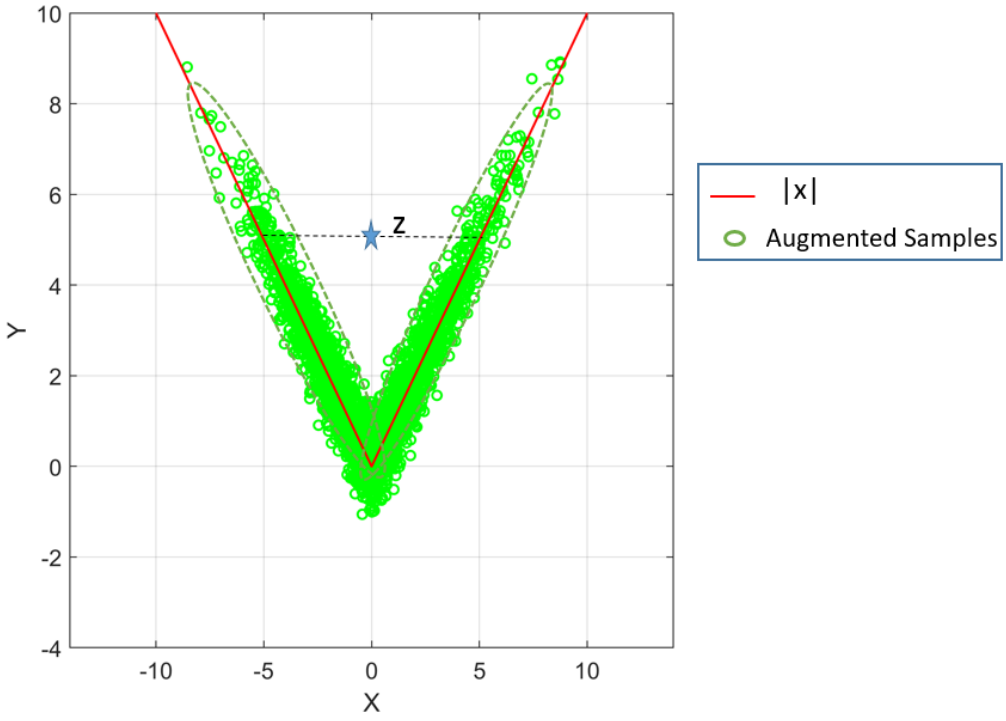


Figure 6.8: Optimal prior density estimate for augmented PGM-I.

6.3 Density Estimation

Density estimation of the propagated PDF has a central role in PGM filtering. In fact, the theoretical guarantees we derived in chapters 4 and 5 were obtained assuming the availability of a perfect density estimator. In chapter 6, we derived an augmented PGM-I filter, which can handle measurement nonlinearity and non-Gaussianity, whose performance is mainly the function of the density estimation algorithm. Clearly the performance of the density estimation algorithm is of great significance to PGM filtering performance. In this chapter, we will explore some aspects of density estimation problem in relation to PGM filtering.

6.4 Number of Modes

Several density estimation algorithms rely on a mixture model representation of the PDF. Some of the most widely used algorithms such as K-means and E-M algorithms require the number of mixture modes to be specified as an input. In this section we shall investigate how the number of mixture modes used influence the PGM estimation performance. We shall consider two examples that are taken from [56].

1. **Quadratic, Univariate Model**: In this example, we use the PGM filters to estimate the state of a one dimensional system. Starting with the initial PDF, we perform a single propagation step followed by a measurement update step. The resulting filtered posterior PDF is then compared with theoretically obtained posterior PDF. The parameters for this example problem are listed below.

   Parameters:

   - Initial distribution: Gaussian with $\mu = 0, P = 20$.

- Markov transition PDF: $f(x) = x, Q = 20$.
- Likelihood Function : $h(x) = \frac{x^2}{20}, R = 50$.
- Observation: $y = 30$.

We use the PGM-1 Filter, augmented PGM-I filter and PGM-II filter to estimate the state of this dynamical system. We use the k-means basde clustering algorithm we developed in chapter 4 to estimate the propagated prior density. The results obtained by filtering this system as the parameter N_{max} is increased from 1 to 8 are plotted in Figure 6.9.

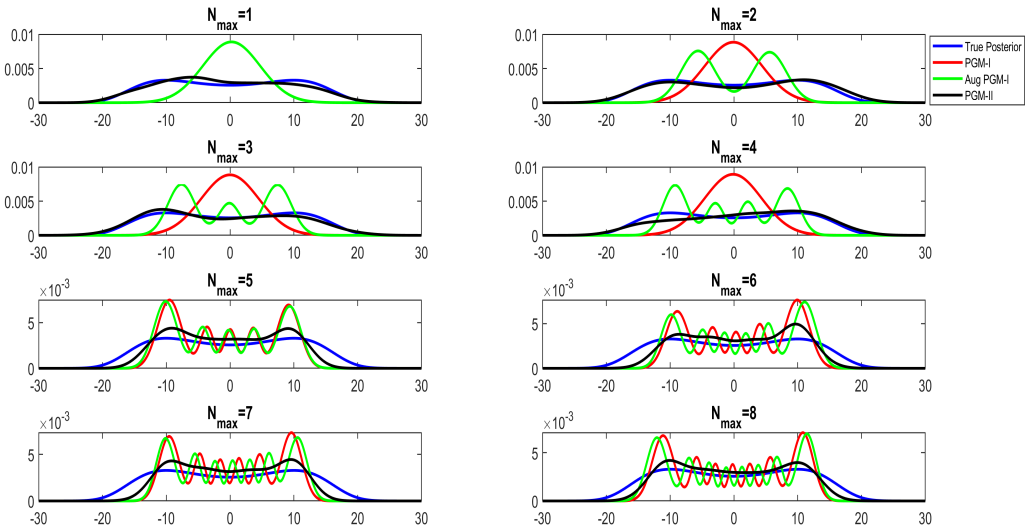


Figure 6.9: Estimated density, Quadratic Univariate Model

2. **Cubic, Univariate Model**: Like the quadratic system of previous example, we perform a single propagation step followed by a measurement update

step. The parameters for the cubic univariate system model are listed below.

Parameters:

- Initial distribution: Gaussian with $\mu = 0, P = 20$.
- Markov transition PDF: $f(x) = x, Q = 20$.
- Likelihood Function : $h(x) = \frac{x^3}{120}, R = 50$.
- Observation: $y = 20$.

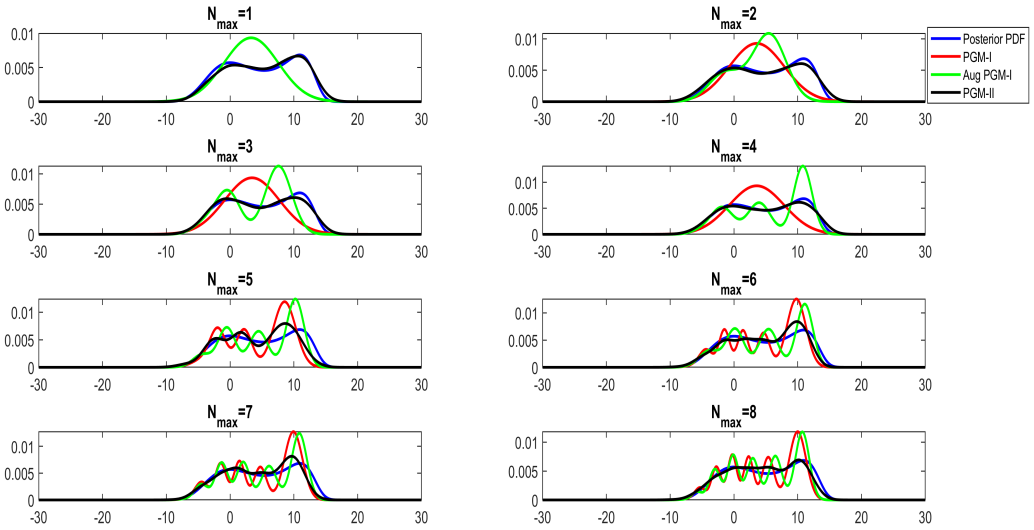


Figure 6.10: Estimated density, Cubic Univariate Model

The filtered posterior PDFs for the cubic univariate model are plotted along with the theoretical posterior PDF in Figure 6.10. Note that, the PGM-II filter only obtains samples from the posterior PDF. What we have plotted as the PGM-II curve in Figures 6.9 and 6.10 is the kernel density estimate obtained from the PGM-II posterior samples. The results indicate that as N_{max}

increases, the filtered posterior PDF obtained by the PGM-I filter and augmented PGM-I filter start to resemble more and more like the true posteior PDF. In fact, as we increase the maximum number of components, the augmented PGM-I filter catches up with the true posterior PDF faster than the original PGM-I filter. However, the differences start to disappear as the total number of modes used becomes much larger than the actual number of posterior modes. The performance improvement of PGM-I filters with the increase in N_{max} is an interesting observation. Clearly the prior PDF in both experiments is unimodal and given the linear dynamics, the ideal density estimator should only be able to identify a single propagated prior component. However, as we have seen from our results, using more components than the ideal can improve PGM-I filtering performance. This happens because the total error in the filtered posterior PDF is a function of both the density estimation error and the linearization error in measurement update. When the total number of components increase beyond the ideal, the density estimation error increases. However, the resultant splitting of the domain of the measurement function will result in a reduction of linearization errors. In PGM-I filters, the gain in accuracy due to latter compensates for the error made in density estimation. The augmented PGM-I filter is able to perform better than the PGM-I filter at lower N_{max} as the former has access to measurement nonlinearity information. Note that the PGM-II filter does not experience a gain in accuracy with N_{max} since it does not rely on linearizations to perform measurement updates. Hence not only does splitting the domain not improve its performance, the errors incurred in estimating the propagated prior can make the posterior estimates worse.

6.5 Kernel Density Estimators

Density estimation algorithms such as k-means and E-M algorithm assume the PDF of the random variable belongs to some family of distributions. For e.g., in our simulations, we often assumed that the PDF of the propagated random variable is actually a Gaussian mixture. Assuming an underlying structure for the PDF allows us to represent it in terms of a set of a parameters whose size does not grow with the sample size. Indeed, it is only with strong assumptions on the structure of the unknown PDF can we reduce the search space for the estimator to the finite dimensional parametric space. However, these assumptions result in estimation errors as the actual underlying PDF need not belong to the assumed class of distributions. Non-parametric density estimators are a class of algorithms that estimate the PDF without assuming any underlying structure. Histograms, Kernel density estimators(KDE) etc are some of the most commonly used non-parametric estimation techniques. Where as histograms are discontinuous at bin boundaries, the KDE offers a continuous non parametric estimator of the PDF. Formally, given a collection of random samples x_i distributed according to the true density $p(x)$, the KDE estimator is computed as

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x - x_i}{h}). \tag{6.44}$$

Here the kernel K is a non negative continuous function that is assumed to integrate to the width parameter h. Gaussian, Epanechnikov, Quartic etc are some of the commonly used kernel functions. We can obtain a KDE based PGM filter by replacing the k-means based GMM estimation step with kernel density estimation. We designed a variant of the PGM-II filter that uses an Epanechnikov KDE of the propagated prior density. The value of Epanechnikov kernel function at a point u on

the real line can be evaluated as

$$K(u) = \frac{3}{4}(1 - u^2), \quad |u| \leq 1. \tag{6.45}$$

When the total number of samples n is high, evaluating $\hat{p}(x)$ can be very expensive. In order to keep the computational cost under control, we run a nearest neighbour algorithm at x and identify the k nearest neighbours from the set of random samples x_i. We approximate $\hat{p}(x)$ with a local KDE made up of the k nearest neighbours of the point x. We tested the performance of the KDE based PGM-II filter against the GMM based PGM-II filter in the Cubic univariate model. However, instead of the univariate Gaussian distribution, we used $\Gamma(x, 2, 2)$ as the initial distribution. The window size for the Epanechnikov kernel is set to be 0.0009 and the number of nearest neighbours is kept at 30. As seen in Figure 6.11, the KDE based PGM-II filter is

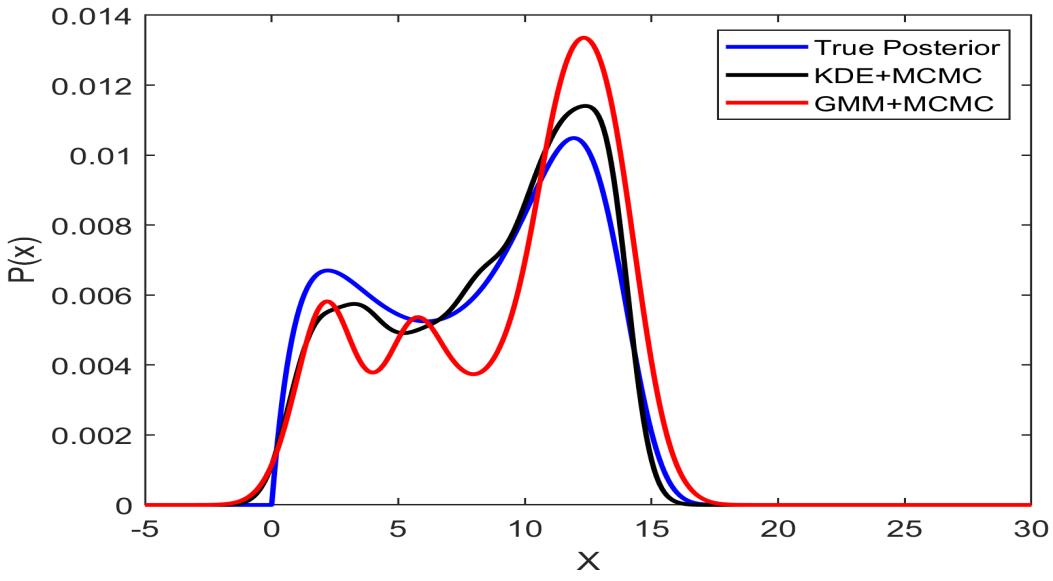


Figure 6.11: Estimated density, Cubic Univariate Model using a KDE based PGM-II Filter

found to perform better than the GMM based filter. However, its performance is found to be sensitive to window size selection.

KDE based approaches are in general more flexible in comparison to parametric density estimators. However, they are known to have slow convergence towards the true density. In fact, a KDE based PGM-II variant was found to diverge in the blind tricyclist problem. Clearly, the choice between parametric and non parametric algorithms has to be exercised based on the nature of the application and available computational resources. The PGM filter design is flexible in that it allows the user to freely choose the density estimation algorithms in this manner. PGM-II filter in particular does not enforce any constraints on the family of distributions or Kernels that can be used to model the PDF.

7. DISTRIBUTED PGM-I FILTER

Decentralized estimation of dynamical systems is an important engineering problems with numerous applications in robotics and control[57]. It involves the estimation of the state of a dynamical system using information collected by a network of several sensors/agents without the help of a central node. The network is assumed to allow bidirectional communication between all agents that are connected (Figure 7.1). The knowledge about the system is updated across the network via multi sensor data fusion.

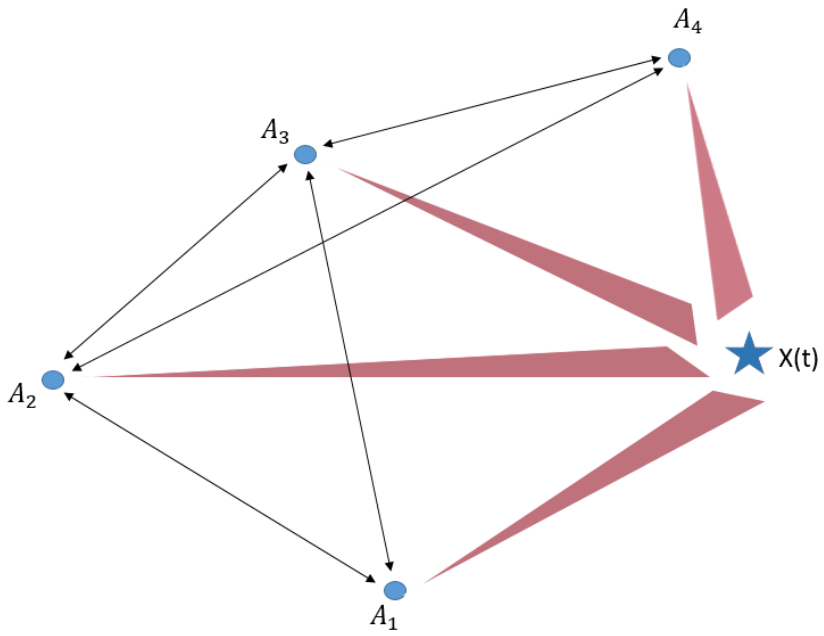


Figure 7.1: Decentralized estimation of the dynamical system $X(t)$ by 4 agents

The information sharing between the agents can take place in several ways. The network connectivity can be used to exchange

1. Measurement models and recorded observations

2. Likelihood functions

3. Individually obtained posterior PDFs

In this chapter, we shall focus on the first option, i.e, the transmission of measurement models and observations across the network. In the most general estimation scenario, the measurement models used by an agent is a nonlinear function of $X(t)$ that may not have a compact parametric representation. This means that the agent may need to transmit the full nonlinear function along with the value of the measurement. The cost of communicating this information is usually high. As discussed in chapter 4, several nonlinear filters such as the PGM-I filter linearizes the nonlinear measurement function as it enables them to obtain the posterior PDF using the simple Kalman measurement update. This has special significance in the decentralized estimation scenario as linearizing the measurement function will also help to keep the cost of communication manageable. In this section, we will discuss a preliminary development towards the design of a decentralized PGM-I filter.

For ease of exposition, we shall consider a scenario that involves only two agents. Let A_1 and A_2 be two agents observing a random dynamical system. Let $X(t)$ represent the state of the dynamical system at time t. Assume that the agents are capable of estimating the state $X(t)$ by generating independent observations z_1 and z_2 and by sharing information between them when connected. Note that even though the development herein is for the two agents scenario, it can be generalized to the n-agent estimation problem in a straightforward fashion. Consider the scenario where the

prior PDFs of the two agents are represented as Gaussian mixture models.

$$P_1(x) = \sum_1^{n_1} \theta_{1,i} \mathcal{N}(\mu_{1,i}, \Sigma_{1,i}) \tag{7.1}$$

$$P_2(x) = \sum_1^{n_2} \theta_{2,i} \mathcal{N}(\mu_{2,i}, \Sigma_{2,i}) \tag{7.2}$$

Note that we have not included time indices in the above as only a one step measurement update is considered here. To simplify the problem further, we shall assume that the agents are connected prior to the measurements arrival so that $P_1(x) = P_2(x) = P(x)$ and $n_1 = n_2 = n, \theta_{1,i} = \theta_{2,i} = \theta_i, \mu_{1,i} = \mu_{2,i} = \mu_i, \Sigma_{1,i} = \Sigma_{2,i} = \Sigma_i$. Then, given the two observations, the posterior PDF of this system is given by

$$\begin{aligned} P(x|z_1, z_2) &= \frac{P(x)P(z_1|x)P(z_2|x)}{\int P(\zeta)P(z_1|\zeta)P(z_2|\zeta)d\zeta} \\ &= \frac{\sum_1^n \theta_i \mathcal{N}(\mu_i, \Sigma_i)P(z_1|x)P(z_2|x)}{\int \sum_1^n \theta_j \mathcal{N}(\mu_j, \Sigma_j)P(z_1|\zeta)P(z_2|\zeta)d\zeta} \end{aligned} \tag{7.3}$$

Let

$$l_i(y) = \int \mathcal{N}(\mu_i, \Sigma_i)P(z_1|\zeta)P(z_2|\zeta)d\zeta \tag{7.4}$$

Then one can multiply and divide the ith term in the numerator by l_i to obtain

$$P(x|z_1, z_2) = \sum_{i=1}^n \frac{\theta_i l_i(y)\mathcal{N}(\mu_i, \Sigma_i)P(z_1|x)P(z_2|x)}{l_i(y)\sum_{j=1}^n \int \theta_j \mathcal{N}(\mu_j, \Sigma_j)P(z_1|\zeta)P(z_2|\zeta)d\zeta} \tag{7.5}$$

Note that the term $\frac{\mathcal{N}(\mu_i,\Sigma_i)P(z_1|x)P(z_2|x)}{l_i(y)}$ represents the component posterior obtained from the Bayesian update of the mixture mode $\mathcal{N}(\mu_i, \Sigma_i)$. We will use $P_i(x|z_1, z_2)$ to denote this PDF. Then we see that the full posterior can be represented as

$$P(x|z_1, z_2) = \frac{\sum_1^n \theta_i l_i(y) P_i(x|z_1, z_2)}{\sum_1^n \theta_j l_j(y)} \tag{7.6}$$

This is a mixture representation of the posterior PDF where the components are given by $P_i^+(x) = P_i(x|z_1, z_2)$ and the weights are given by

$$\theta_i^+ = \frac{\theta_i l_i(y)}{\sum_1^n \theta_j l_j(y)} \tag{7.7}$$

7.1 Obtaining the component posterior PDF

When the measurement model is linearized and the measurement noise assumed is Gaussian, the component posterior $P_i(x|z_1, z_2)$ can obtained using the usual Kalman update. Given the above structure, it is straightforward to obtain a distributed analog to the measurement update. Observe that the measurement update can be broken down into a Bayesian update of a continuous density (to obtain $P_i(x|z_1, z_2)$ from $P_i(x)$) and the Bayesian update of a discrete hidden Markov model (to obtain θ_i^+ from θ_i).

Let y_i and Y_i be defined as

$$y_i = \Sigma_i^{-1}\mu_i, \tag{7.8}$$

$$Y_i = \Sigma_i^{-1}. \tag{7.9}$$

for $i = 1, \cdots n$. The terms y_i and Y_i, known as the information vector and information matrix respectively, encode the information contained in the i^{th} prior PDF component. Given the measurements z_1 and z_2, the information vector and matrix

for the posterior component (y_i^+, Y_i^+) can be then obtained as

$$y_i^+ = y_i + \delta i_1 + \delta i_2, \tag{7.10}$$

$$Y_i^+ = Y_i + \delta I_1 + \delta I_2. \tag{7.11}$$

The above equations represent the Kalman measurement update in the information form. The terms $\delta i_j, \delta I_j$ can be calculated readily from each agent as $\delta i_j = H_j' R_j^{-1} z_j$ and $\delta I_j = H_j' R_j^{-1} H_j$. Once the posterior information vector and matrix for each component is computed, it is straightforward to obtain the corresponding mean and covariance. However only the agent A_j has direct access to $(\delta i_j, \delta I_j)$. By running a standard consensus algorithm such as distributed averaging [58] over $(\delta i_j, \delta I_j)$ across the network, all agents can perform Kalman measurement updates on all local mixture components.

Let the nonlinear measurement functions be

$$z_1 = h_1(X) + \nu_1, \tag{7.12}$$

$$z_2 = h_2(X) + \nu_2. \tag{7.13}$$

At first glance, the PGM-I update equations given in chapter 4 do not appear to be in the information form as given in equations 7.10, 7.11. However, it can be shown that [8] the PGM-I update using z_1 corresponds to linearizing the measurement function $h_1(x)$ through a statistical linear regression and then performing the information update. For e.g. the PGM-I UT corresponds to obtaining a linear regression approximation of the nonlinear measurement function through the sigma points. Given a single agent A_1, a single measurement z_1 and a prior PDF with n components, the PGM-I filter performs n separate Kalman measurement updates. These n updates

correspond to n different linearizations. Consequently, in the multi-agent scenario that we consider here, we will first need to obtain the linearizations of the functions h_1, h_2 over all n prior components $P_i(X)$. Each one of these $2n$ linearizations will then need to be communicated over the network. That is, the actual decentralized PGM-I update will have the form

$$y_k^+ = y_k + \delta i_1^k + \delta i_2^k, \tag{7.14}$$

$$Y_k^+ = Y_k + \delta I_1^k + \delta I_2^k. \tag{7.15}$$

with $\delta i_j^k = H_j^{k'} R_j^{-1} z_j$ and $\delta I_j^k = H_j^{k'} R_j^{-1} H_j^k$, where H_j^k is the linearization of the function $h_j(x)$ at the k^{th} component prior $P_k(x)$ (Figure 7.2).

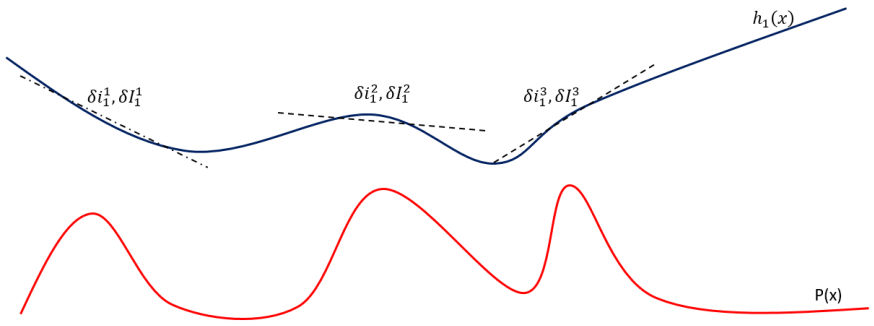


Figure 7.2: Three separate linearizations of the function $h_1(x)$ corresponding to three prior PDF components

7.2 Obtaining the weights

Clearly, obtaining the posterior weights require evaluating the term $l_i(y)$. From Bayes theorem we have

$$P_i(x|z_1, z_2) = \frac{P_i(x)P(z_1|x)P(z_2|x)}{l_i(y)}. \tag{7.16}$$

So once we have evaluated $P_i(x|z_1, z_2)$ we can substitute $x = x_c$ in the above equation to get

$$l_i(y) = \frac{P_i(x_c)P(z_1|x_c)P(z_2|x_c)}{P_i(x_c|z_1, z_2)}, \tag{7.17}$$

where x_c is some constant. In the linear Gaussian case, the component posterior $P_i(x|z_1, z_2)$ can be obtained at each agent by running consensus over the information vectors and matrices. So we assume that $P_i(x_c)$ and$P_i(x_c|z_1, z_2$ are available to both agents. This leaves us with the product in the numerator $P(z_1|x_c)P(z_2|x_c)$. Agent 1 has only access to $P(z_1|x_c)$ and Agent 2 only to $P(z_2|x_c)$. However the product $P(z_1|x_c)P(z_2|x_c)$ is independent of mode i. To see this, substitute Eq.10 in Eq.8.

$$\begin{aligned}\theta_i^+ &= \frac{\theta_i \frac{P_i(x_c)P(z_1|x_c)P(z_2|x_c)}{P_i(x_c|z_1,z_2)}}{\sum_1^n \theta_j \frac{P_j(x_c)P(z_1|x_c)P(z_2|x_c)}{P_j(x_c|z_1,z_2)}} \\ &= \frac{\theta_i \frac{P_i(x_c)}{P_i(x_c|z_1,z_2)}}{\sum_1^n \theta_j \frac{P_j(x_c)}{P_j(x_c|z_1,z_2)}}.\end{aligned} \tag{7.18}$$

This means that once we have access to the component posterior $P_j(x|z_1, z_2)$, we can evaluate a number proportional to $l_i(y)$ by merely substituting an x_c in the prior and posterior components.

We test the performance of the distributed averaging approach with the following target tracking example. We consider a grid of 4 sensors placed at $(0, 0), (0, 4), (4, 0)$

and $(4, 4)$. The true target position and the sensor locations are visualized in Figure 7.3

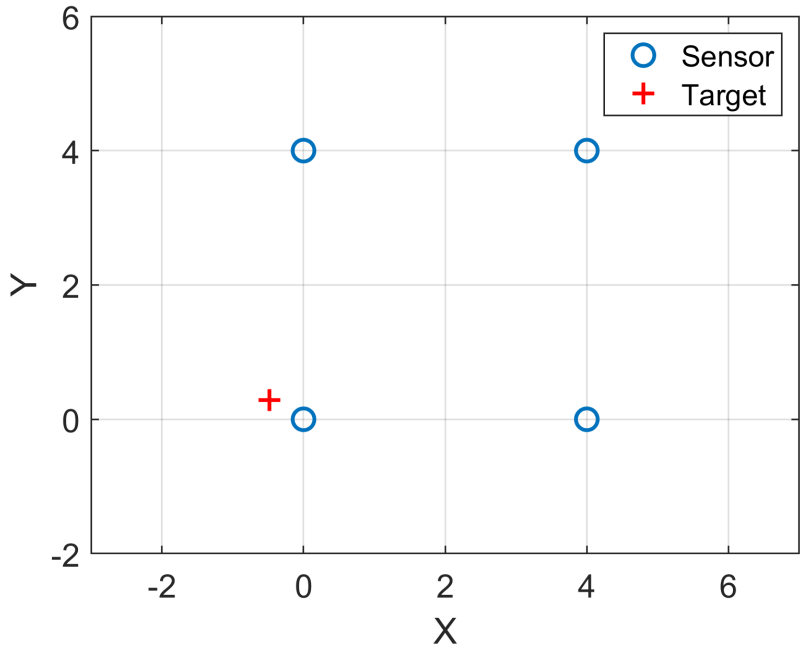


Figure 7.3: Decentralized target tracking, Set up

Together they track a target that is capable of moving on the $x-y$ plane. In this example we consider the PGM performance in a single decentralized measurement update. The prior PDF for the target is given by

$$P(x) = 0.3\mathcal{N}(\mu_1, \Sigma_1) + 0.4\mathcal{N}(\mu_2, \Sigma_2) + 0.3\mathcal{N}(\mu_3, \Sigma_3), \tag{7.19}$$

where $\mu_1 = \begin{bmatrix} -1 & 1 \end{bmatrix}, \mu_2 = \begin{bmatrix} 1 & 2 \end{bmatrix}, \mu_3 = \begin{bmatrix} 2 & 3 \end{bmatrix}$ and $\Sigma_1 = \mathbb{I}_{2\times 2}$, $\Sigma_2 = 1.2\mathbb{I}_{2\times 2}$, $\Sigma_3 = 2\mathbb{I}_{2\times 2}$. Each sensor records the distance between the sensor and the target

according to

$$z_i = ||x - \rho_i||_2 + \nu_i, \tag{7.20}$$

where ρ_i is the location of the i^{th} sensor. The measurement noise variances is $R_i = 0.4$ for $i = 1....4$. Once the measurements are recorded, the information vectors and matrices are summed up via distributed averaging and a Kalman measurement update is performed using equations 7.14, 7.15. The posterior PDF obtained using decentralized PGM-I update is visualized in ensemble form in Figure 7.4. Two of the three modes are seen to survive and the majority of the probability is found to be located within the mode that contains the true target position.

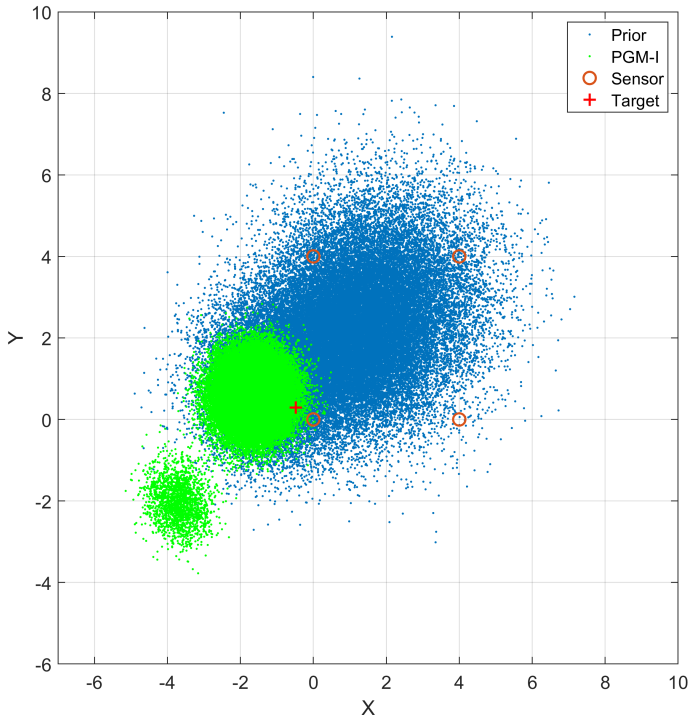


Figure 7.4: Decentralized target tracking, Prior and Posterior PDF

8. PARALLEL MARKOV CHAIN MONTE CARLO*

The MCMC methods are a class of Markov chain based sampling algorithms that are widely used to obtain samples from probability densities that are normally difficult to sample from. They obtain samples by generating a reversible Markov chain whose equilibrium density coincides with the target density $P(X)$. Lets consider the Metropolis-Hastings algorithm, a widely used variant of MCMC, that relies on a proposal density $Q(x|y)$ to generate samples from the target density $P(X)$. Given the target density and the proposal density, it produces a sequence of Markov chain transitions, each consisting of a sampling and an accept/reject operation. Let x^{t-1} be the state of the chain at $t-1$, then the Metropolis-Hastings algorithm first samples a new candidate point x^{*t} from the proposal density $Q(x|x^{t-1})$. The point x^{*t} is chosen as the next state of the Markov chain with an acceptance probability α where $\alpha = min\{1, \frac{P(\underline{x}^{*\underline{t}})Q(\underline{x}^{\underline{t}-\underline{1}}|\underline{x}^{*\underline{t}})}{P(\underline{x}^{\underline{t}-\underline{1}})Q(\underline{x}^{*\underline{t}}|\underline{x}^{\underline{t}-\underline{1}})}\}$. The sampling and acceptance operations, when combined gives rise to a transition kernel whose steady state density equals the target PDF $P(x)$,i.e., as $t \to \infty$ the distribution P_t of the point X_t will converge to the target density $P(x)$. This has made MCMC methods invaluable for sampling from distributions that are otherwise difficult to sample from.

In practice, running the simulation until steady state will not be possible. Instead the chain is run long enough until the distribution P_t is sufficiently close to the true distribution $P(x)$. All samples obtained before a burn in time $T_{burn-in}$ are discarded. The time taken by the Metropolis Hastings chain to ensure $||P_t(x) - P(x)|| < \epsilon$ for a given distance measure and error bound ϵ will depend on both the proposal density $Q(x)$ and the target PDF $P(X)$. In this sense, a Markov chain that converges quickly

*Parts of this chapter were reprinted with permission from [7].

to the target distribution is called quickly mixing. In practice, Markov Chains mix much more quickly when the target distribution is not multi-modal. For multi-modal target distributions, the time taken to satisfy a given error bound can be impractically large. In the multi modal scenario, the chain will require a really large time to explore the state space effectively since moving from one mode to another may require a large jump which happens only with a small probability. However, if we could overcome the inherently sequential nature of the Metropolis Hastings algorithm, we may be able to explore multiple regions of the state space simultaneously. To this end, parallelizable MCMC methods that perform asynchronous sampling over the state space to compute integrals [47], [59] and perform Bayesian estimation [4] have been proposed recently. It is to be noted that there is also strong incentive to parallelizing MCMC methods as serial CPU speed ups have plateaued over recent years and the prevailing computing paradigm has shifted to parallel processing[47].

To demonstrate the effectiveness of running several Markov chains in parallel, we conduct the following experiment. We attempt to sample from a multimodal PDF using the Metropolis Hastings algorithm. Our target PDF $P(x)$ is a Gaussian mixture with four equally weighted components. The four components are placed along the Cartesian coordinate axes at points $(r,0),(0,r),(-r,0),(0-r)$ respectively. Components 1 and 3 have their semi major axis along the X-axis. The semi-major axis for the components 2 and 4 are placed along the Y-axis. The covariance of component 1 is given by

$$C_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.25 \end{bmatrix}. \tag{8.1}$$

Covariances for components 2,3 and 4 are given by rotating C_1 by $\frac{\pi}{2}, \pi$and $3\frac{\pi}{2}$ radians respectively. The target PDF is represented in ensemble form in Figure 8.1.

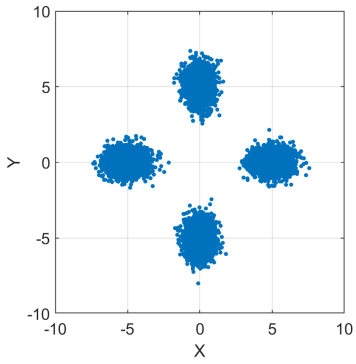


Figure 8.1: Four component Gaussian mixture target PDF

Now we shall use the Metropolis Hastings algorithm to sample from this PDF. Starting from the origin, we will use a random walk with proposal covariance C_Q

$$C_Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{8.2}$$

We collect 4000 samples after burn in of 1000 samples. The results are shown in Figure 8.2. We see that the chain only sampled from component 1.

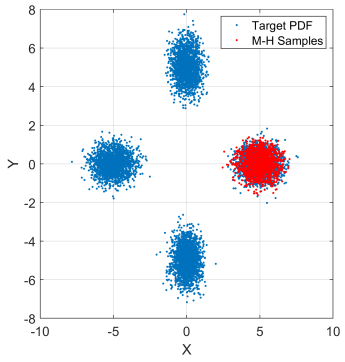


Figure 8.2: Target PDF and the M-H samples;Test1

To investigate whether this is a result of insufficient burn-in or simply insufficient number of total samples, we increased the the burn in time to 5000 and then collected 8000 samples.The experiment is repeated four times. The results are shown in Figure 8.3.

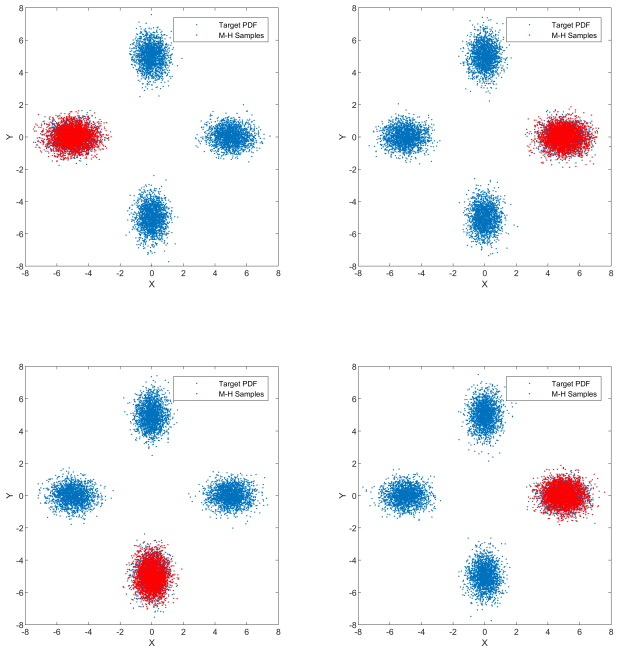


Figure 8.3: Target PDF and the M-H samples from 4 experiments; After increasing burn-in to 4000 and collecting 8000 samples

Despite increasing the burn in period and the total number of samples, the M-H algorithm is found to sample from only one of the four mixture components in any given experiment. This shows that spending more computational resources in a sequential manner may not produce a comparable reduction in the error $||P_t(x) - P(x)||$. Now we will conduct the same experiment by running 2 parallel chains. Each chain is

allocated 3000 points that can be sampled after a burn in of 1000. The total number of samples obtained in this case will be 6000 which is far less than the 13000 we used in the previous experiment. The results are displayed in Figure 8.4. The two parallel chains are seen to converge to two separate modes in three out of the four tests.

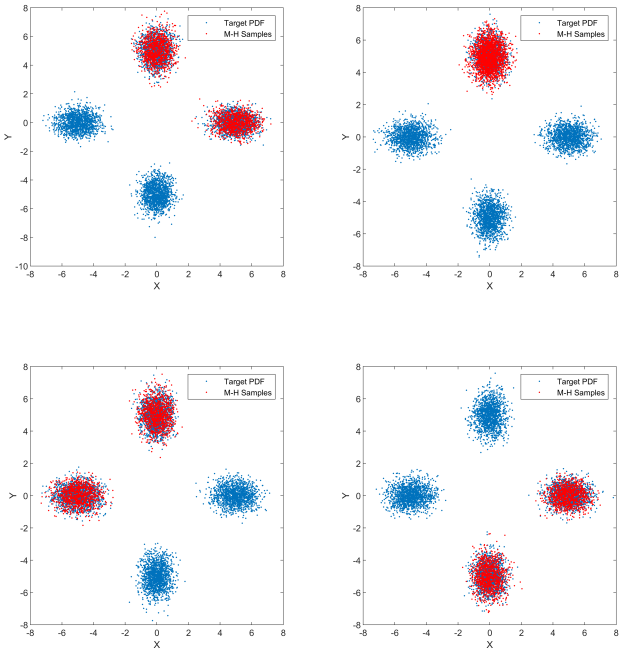


Figure 8.4: Target PDF and the M-H samples from 4 experiments; Running two chains in parallel with burn-in of 1000 and collecting 3000 samples

Next we increase the number of parallel chains to four and the results are plotted in Figure 8.5. The sampling performance is seen to have improved as three modes were captured in three out of 4 tests. In tests with two and four parallel chains, we observe that the number of modes captured in sampling is sometimes less than

the number of chains. This is due to multiple M-H chains converging to the same mixture component.

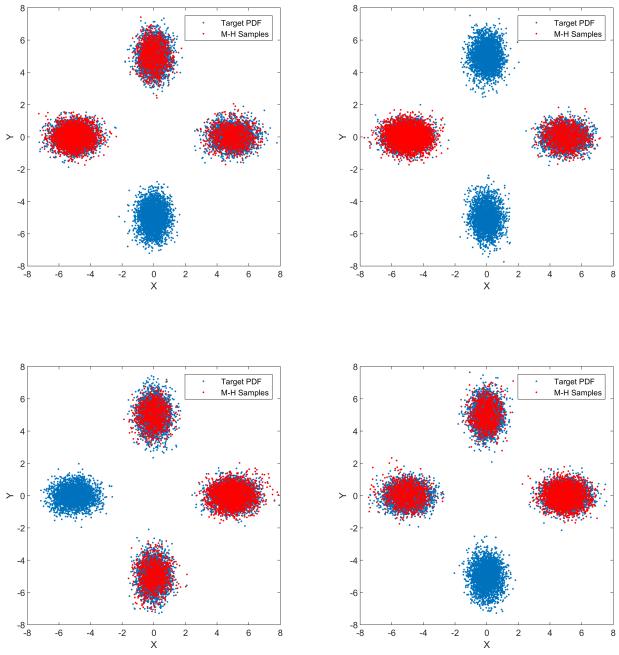


Figure 8.5: Target PDF and the M-H samples from 4 experiments; Running four chains in parallel with burn-in of 1000 and collecting 3000 samples

Since the probability of this happening is non-zero, in practice we may need more than M parallel chains to capture all modes of an M component Gaussian mixture. We increase the number of parallel chains to six and find that the chains find all four modes of the target PDF in two test cases. The results are shown in in Figure 8.6.

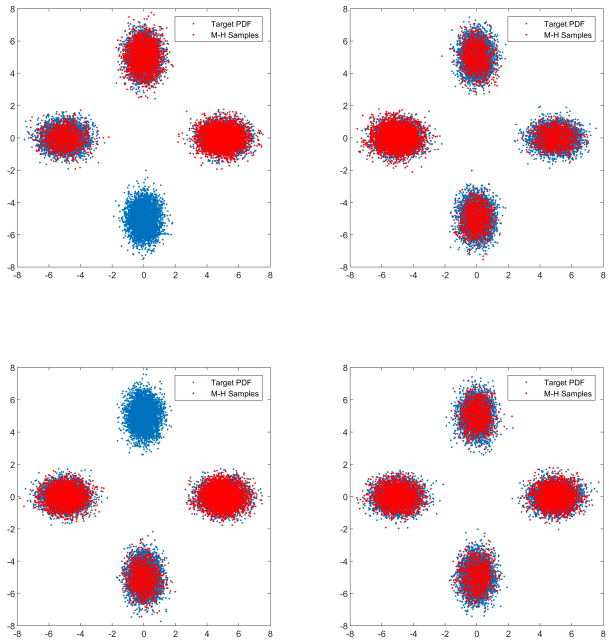


Figure 8.6: Target PDF and the M-H samples from 4 experiments; Running six chains in parallel with burn-in of 1000 and collecting 3000 samples

A pseudo code description of the complete parallel MCMC algorithm is given in Algorithm 5. The cluster weights ω_k in Algorithm 5, step 7 can be evaluated as

$$\omega_k = \int_{S_k} P(x)dx. \tag{8.3}$$

Here S_k is a subset of the state space such that $A_k \subseteq S_k,\ S_{k_1} \cup S_{k_2} = \phi$ for $k_1 \neq k_2$. In practice the above integral is approximated with a discrete sum evaluated using samples from A_k. Our experiments clearly reveal that it is advisable to run multiple parallel M-H chains than expend more computational resources sequentially when trying to sample from a multi modal target PDF. Additionally, If each chain is only

Algorithm 5 Parallel MCMC Algorithm

Given target PDF $P(x)$, proposal density $Q(x|x^{t-1})$, initialization density $g(X)$, number of modes M and number of parallel chains n_{par}

1. **for** $j = 1 \rightarrow n_{par}$ do
2. Obtain the initial sample X_j^0 from $g(X)$
3. Use MCMC to sample $X_j^{(i)}, 0 < i \leq N$, from $P(x)$ using the proposal density $Q(x|x^{t-1})$
4. **end for**
5. Evaluate the set S of all sampled points, $\Phi = \bigcup_{j=1}^{n_{par}} \Phi_j$ where $\Phi_j = \{X_j^0 \cdots, X_j^N\}$
6. Use a clustering Algorithm $\mathcal{C}$ to separate the set Φ into M distinct clusters A_k, $k = 1 \cdots M$
7. Evaluate the cluster weights $\omega_k,\ k = 1 \cdots M$
8. Sample N particles from the weighted collection of ensembles $\{(\omega_k, A_k)\}$

able to capture one mode, then we may need to run at least as many chains as there are modes to the target PDF.

8.1 Application to Optimization

The applications of MCMC methods is not just limited to sampling. Given a general high dimensional PDF P(X), the Markov chain constructed using the MCMC will efficiently explore the domain of X and locate itself in regions of high probability. This characteristic of MCMC methods can be utilized to perform optimization of a function $g(x)$ over the search space X. Simulated annealing (SA) is a probabilistic optimization algorithm that uses this Markov chain construction to perform global optimization [60]. The target density for minimizing the cost function $g(x)$ using the SA algorithm is constructed as $P(x) = Cexp(-g(x)/T)$ where C is a normalizing

constant. The term T is a temperature parameter that, starting from sufficiently high initial value T_0, has to be gradually cooled down. The random jumps involved in the Markov chain simulation helps the SA algorithm to escape local minima. However, the optimization performance of the SA algorithm depends strongly on the rate at which the cooling is administered. Furthermore, like Metropolis Hastings, it is an inherently sequential algorithm. The parallel MCMC (p-MCMC) method can be readily used a sanpling based optimization procedure to find the global minimum of a function $g(x)$ with multiple local minima. The p-MCMC optimization method is similar to the multi-start heuristics that are widely used in global optimization problems. It runs multiple Markov chains, each starting from a different initial state $x_i(0)$, in parallel to explore the state space, aspiring to find the global optimum of the objective function $g(x)$. The target density used by the p-MCMC method is similar to that used by the SA algorithm. However, unlike the SA algorithm, the temperature parameter T will be kept constant through out the simulation. As a result, the p-MCMC chains will be time homogeneous. Depending on the value of the temperature parameter T, individual chains may converge to some optimal state in their neighbourhood. However, given that the target function $g(x)$ has finite number of local optima, the probability that at least one p-MCMC chain will sample the global optima will converge to 1 as the number of parallel chains increases. Next we will test the performance of the p-MCMC method in a few optimization problems.

8.1.1 Fifth De Jong Function

The fifth De Jong function (DJ5) is a two dimensional optimization function with several local minima. It is widely used to test the performance of global optimization

algorithms[61, 62]. The DJ5 function is defined as

$$f(x_1, x_2) = \left(\frac{1}{500} + \sum_{1}^{25} \frac{1}{i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6} \right)^{-1} \tag{8.4}$$

$$a = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix} \tag{8.5}$$

The search space for finding the optima is usually limited to the square given by $x_i \in \left[-65.536, 65.536\right]$, $i = 1, 2$. The DJ5 function is plotted in Figure 8.7.

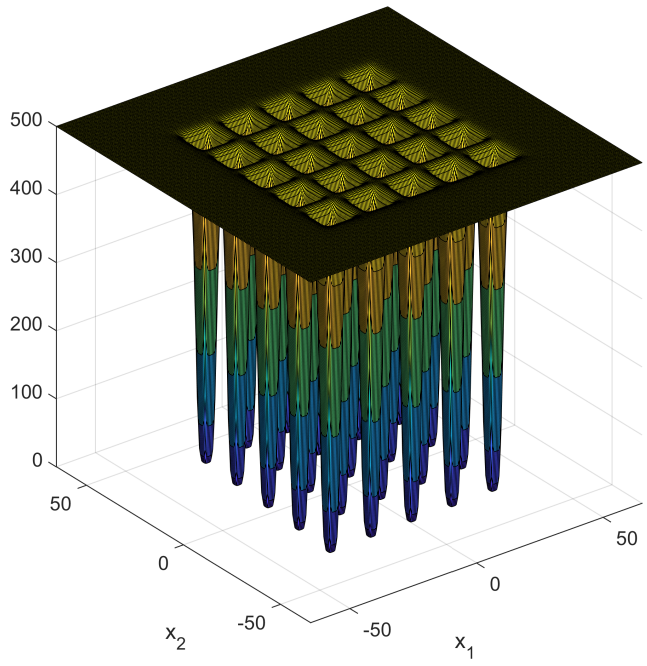


Figure 8.7: Fifth De Jong Function

The global minima for the DJ5 optimization problem is located at $(-32, -32)$ with $f(-32, -32) \approx 0.998004$ [62]. We will attempt to minimize this function using the

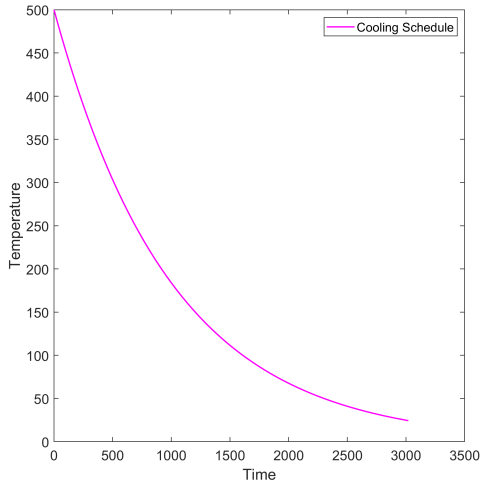


Figure 8.8: Cooling schedule for the simulated annealing algorithm for DJ5 function optimization

p-MCMC method and compare its performance with that of simulated annealing algorithm. We will use 10 parallel M-H chains each sampling 3000 points after a burn-in of 20 points. For simulated annealing we used a single M-H chain with a cooling schedule of the form $T_n = T_0\delta^n$ with $\delta < 1$. For our simulations, we used $\delta = 0.999$. The cooling schedule is plotted in Figure 8.8. The M-H chain for the simulated annealing algorithm is also allowed to sample 3000 points after a burn in of 20 points.

The samples obtained by the two algorithms are plotted in Figure 8.9. Note that even though the p-MCMC method obtains 10 times as many samples as the simulated annealing algorithm, the time taken will be similar when the speed up due to parallel processing is linear. The minima obtained by the two algorithms are shown in Figure 8.10.

We see that the minimum obtained by the p-MCMC method is much closer to the global minimum. The evolution of the DJ5 cost function along with the

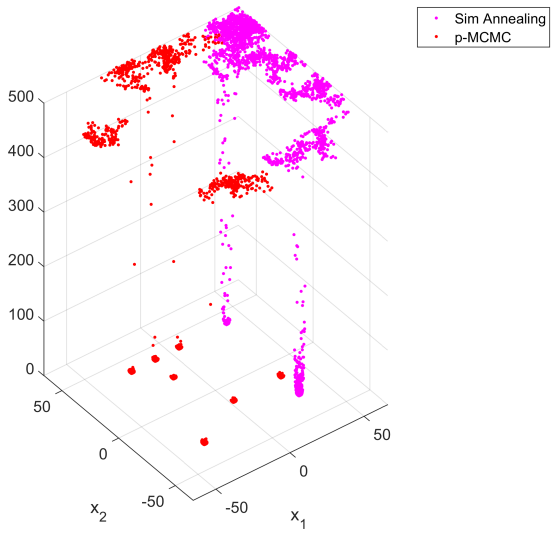


Figure 8.9: Samples obtained from optimizing DJ5 function with p-MCMC method and simulated annealing

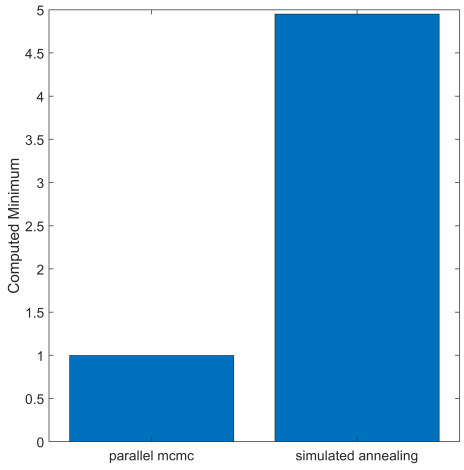


Figure 8.10: Minima of DJ5 function computed by p-MCMC method and simulated annealing

evolution of the Markov chains for the two algorithms is plotted in Figure 8.11. Each parallel Markov chain is observed to quickly fall into the neighbourhood of a minimum where it remains for the rest of the simulation. Since both p-MCMC and simulated annealing are random algorithms, We repeated the simulation 20 times and plotted the minima obtained by the two algorithms in Figure 8.12. The p-MCMC method is seen to outperform the simulated annealing algorithm in majority of the simulations. We repeated the 20 simulation test after increasing the total samples used in the simulated annealing algorithm to 30,000. The results are plotted in Figure 8.13. The p-MCMC method is seen to outperform the simulated annealing even after allowing the latter to sample an equal number of total samples.

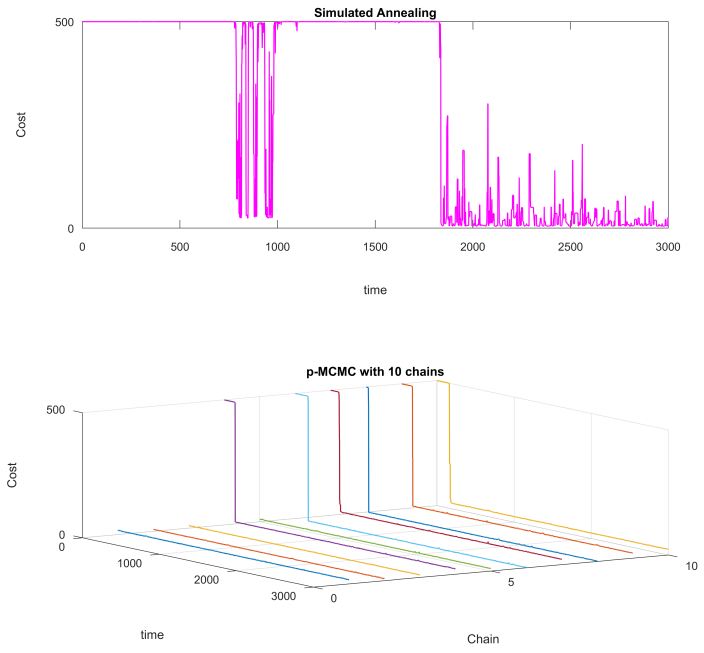


Figure 8.11: Evolution of the DJ5 cost function with the Markov chain

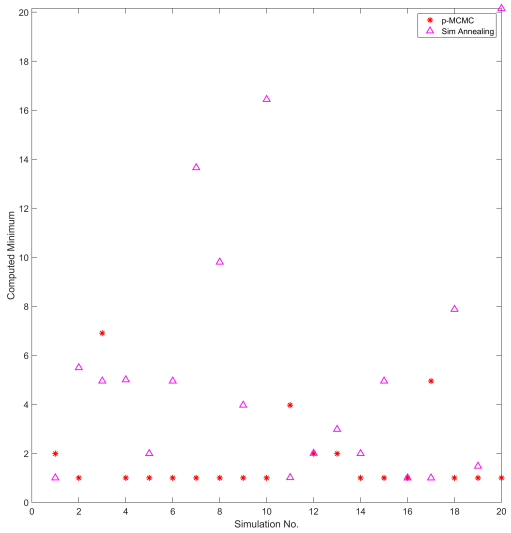


Figure 8.12: Performance of p-MCMC method and simulated annealing over 20 DJ5 optimization runs

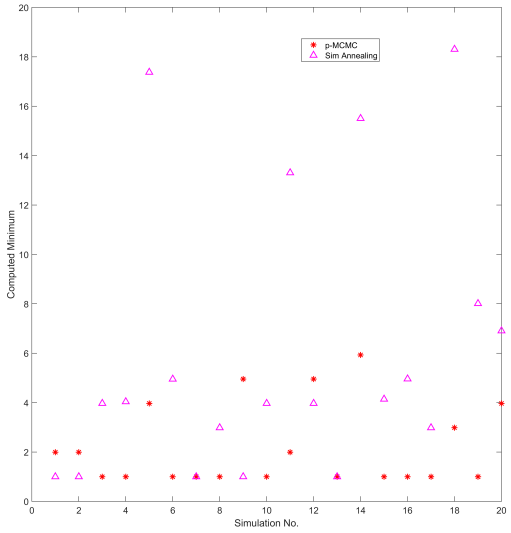


Figure 8.13: Performance of p-MCMC method and simulated annealing over 20 simulations, after increasing annealed samples to 30,000

8.1.2 Six-hump camel back function

The six hump camel back (6HCB) function is a two dimensional function that is widely used to test the performance of global optimization algorithms. The 6HCB function is defined as follows.

$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$
$$-3 \leq x_1 \leq 3, -2 \leq x_2 \leq x_2 \tag{8.6}$$

It has six local minima. Located at $(-0.0898, 0.7126)$ and$(0.0898, -0.7126)$ are the two global minima at which $f(x) = -1.0316$ [63]. The 6HCB function is plotted in Figure 8.14. We will use the p-MCMC method and simulated annealing algorithm to

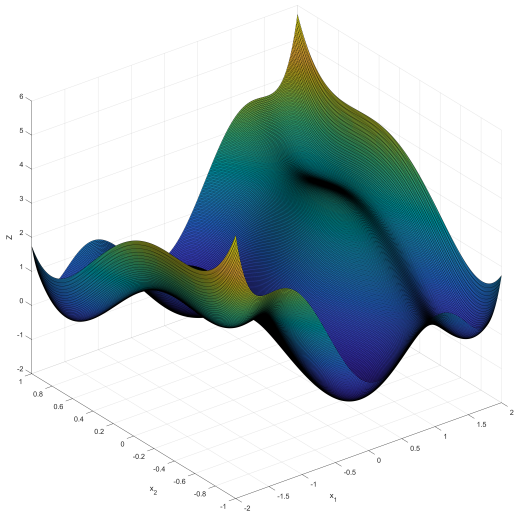


Figure 8.14: Six-hump camel back function

minimize the 6HCB function. The p-MCMC algorithm is implemented with 10 par-

allel M-H chains each sampling 3000 points after a burn-in of 20 points. The cooling schedule used for the simulated annealing algorithm is plotted in Figure 8.15.

The samples obtained by the p-MCMC method and simulated annealing algo-

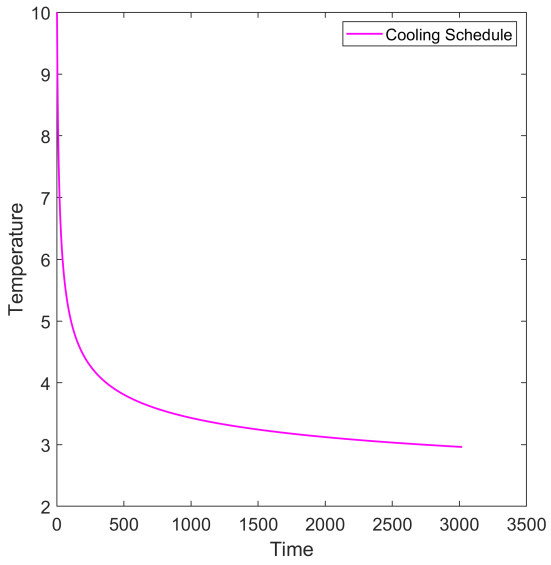


Figure 8.15: Cooling schedule for the simulated annealing algorithm for 6HCB function optimization

rithm are plotted in Figure 8.16. The minima computed by the two algorithms is plotted in Figure 8.17. The two algorithms are seen to offer comparable performance. A similar trend is observed when the experiment is repeated 20 times (Figure.8.18).

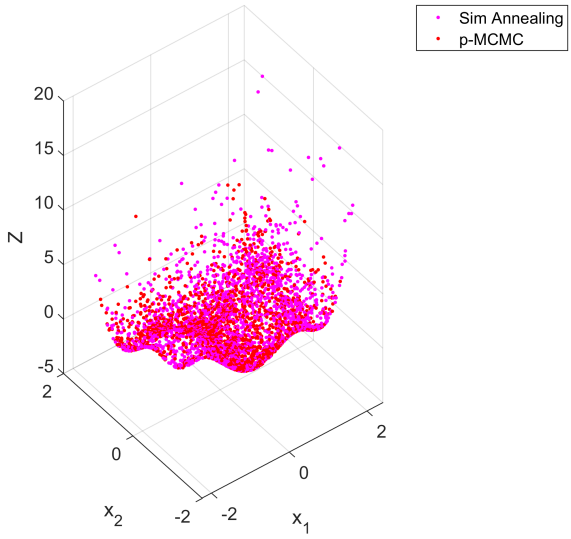


Figure 8.16: Samples obtained from optimizing 6HCB function with p-MCMC method and simulated annealing

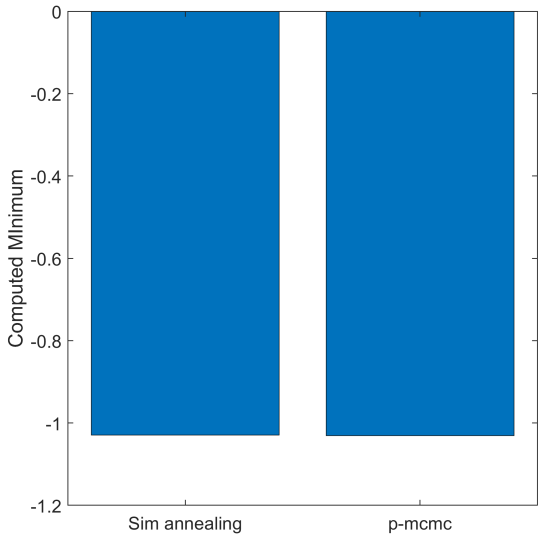


Figure 8.17: Minima of 6HCB function computed by p-MCMC method and simulated annealing

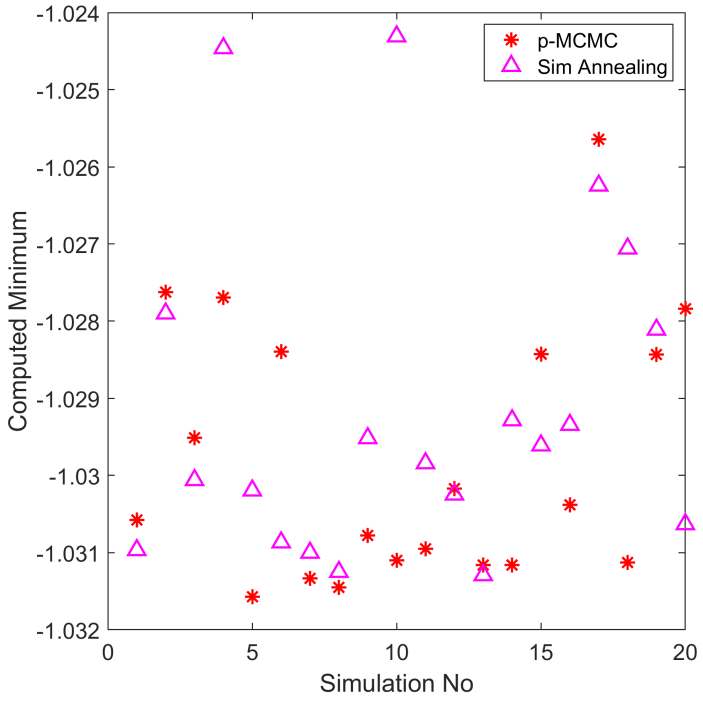


Figure 8.18: Performance of p-MCMC method and simulated annealing over 20 6HCB optimization runs

8.1.3 Sensor Scheduling

Optimal sensor scheduling is a widely studied problem with applications in numerous domains including space situational awareness (SSA). In the context of space surveillance and tracking, the objective of sensor scheduling is to obtain a sensor management strategy that maximizes the information gain from observing a large number of space-based targets using a limited number of sensors. A brief mathematical description of the sensor tasking problem is given below.

Consider a multi target tracking scenario that involves a set of N objects denoted by $O = \{o_1, o_2, \cdots, o_N\}$. Let the state of these objects be represented by $X = \{x_1, x_2, \cdots, x_N\}$. The dynamics of each target is assumed to be governed by a

stochastic differential equation of the following form

$$\dot{x}_i = f(x_i) + w_i. \tag{8.7}$$

Here f represents a suitably accurate orbital motion model and w_i is a white noise process. Assume a set of M sensors, $S = \{s_1, s_2 \cdots s_M\}$, that are capable of recording measurements of the targets O. Each sensor is assumed to be capable of recording observations from within a limited field of vision (FOV) around its current position/look direction. The range of possible look directions is limited by the Field of Regard (FOR) of the sensor. Observations are recorded according to

$$z_k(t) = H_j(x_{j_i}(t)) + \nu_j. \tag{8.8}$$

Here, z_k, $k = 1 \cdots K$, $K \leq N$ represent the observations recorded at time t , H_j represents the measurement function of the sensor s_j and j_i represents the index of the i^{th} object within the FOV of sensor s_j. The measurement noise term ν_j is assumed to be a white noise process. Let $z(t) = \{z_1(t) \cdots z_k(t)\}$ and $Z(t) = \{z(1), \cdots z(t)\}$. Then the posterior probability density function (PDF) of the state of the system at time t is given by the conditional density $P(X(t)|Z(t))$. Given the observations $z(t)$ and the propagated prior PDF $P(X(t)|Z(t-1))$, the PDF $P(X(t)|Z(t))$ can be arrived at using a multi-target tracking algorithm. We shall call this PDF the information state $\chi(t)$ of the system at time t. Clearly, the information state at t is a function of the measurement sequence $z(t)$. Let $\Delta I_z(\chi(t), z(t))$ represent the information gain from updating the predicted state PDF using measurements $z(t)$ at time t. The one step information gain $\Delta I_z(\chi(t), z(t))$ can be quantified in terms of various functions of the information state of the system at t, such as the information entropy,

Fisher information, covariance etc. Note that the measurement sequence $z(t)$ is a random vector that in turn depends on the configuration of the sensors $V(t)$ at time t. Consequently, given a sensor configuration $V(t)$ we can compute the expected information gain as the weighted integral of $\Delta I_z(\chi(t), z(t))$ with $P(z(t)|\chi(t), V(t))$. Let $\Delta I(\chi(t), V(t))$ represent the expected information gain from choosing the sensor configuration $V(t)$ at time t. Then, the objective of the sensor scheduling problem is to obtain the sequence of sensor configurations $V(t_0), V(t_1) \cdots$ that maximize the net information gain from the system. The resulting optimization problem, in its most general form, is to be solved over an infinite horizon, with the reward function given by

$$J(\chi(0)) = \sum_{t=0}^{\infty} \Delta I(\chi(t), V(t)). \tag{8.9}$$

Obtaining the optimal configuration sequence $V(t_0)^*, V(t_1)^* \cdots$ that maximizes the infinite horizon reward in equation 8.9 over all possible sequence of information states is extremely difficult. Here, we consider the optimal scheduling problem for the single sensor multi-target tracking scenario. For each target, we represent the one step information gain $\Delta I_{z_k}(\chi_{S(k)}(t), z_k(t))$ in terms of the incremental change in the determinant of the inverse of it's covariance matrix. Inverse of the covariance of a random vector is also known as its information matrix. Here $S(k)$ is a function that maps the index k the measurement z_k to the index i of the target x_i from which it originated. If $C_{S(k)}(t^-)$ is the covariance of the PDF of target $S(k)$ before measurement z_k and $C_{S(k)}(t)$ the covariance after the measurement, the one step information gain $\Delta I_{z_k}(\chi_{S(k)}(t), z_k(t))$ is given by $\Delta I_{z_k}(\chi_{S(k)}(t), z_k(t)) = |inv(C_{S(k)}(t)| - |inv(C_{S(k)})(t^-)|$ where $|\ |$ is the determinant. The incremental information gain for the full set of targets at time t is defined as the sum of information

gains of individual targets t.

$$\Delta I_z(\chi(t), z(t)) = \sum_{k=1}^{K} \Delta I_{z_k}(\chi_{S(k)}(t), z_k(t)) \tag{8.10}$$

In this example, we consider the problem of obtaining the optimal action sequence that maximizes information gain over a receding horizon of length Δ_t using the p-MCMC algorithm.The p-MCMC method is used to obtain the optimal sensor configuration for the single sensor Multi-target scheduling problem as described below. Let the total number of parallel simulations run at the scheduling instant t be $N_{par}(t)$. Then the p-MCMC algorithm starts by sampling N_{par} different initial sensor configurations $V^0(t) = \{V_1^0, \cdots V_{N_{par}}^0\}$ where each configuration state V_i^0 represents a full sequence of look directions from time t to $t+\Delta_t$. Each sample $V_i(0)$ is used to initialize an MCMC chain whose target density is a function of the expected information gain over a Δ_t window.

The evaluation of expected information gain for the configuration $V_i(m)$ over the space of all possible measurement sequences in the interval from t to $t + \Delta_t$ is computationally tedious. Instead we will obtain an approximate expected information gain $\bar{I}_{\Delta_t}(t, V_i(m))$ using the measurement sequence generated from the mean system path. The approximate expected information gain for the configuration $V_i(m)$ is computed as follows. Given the propagated prior PDF $P(X(t)|Z(t-1))$, we obtain a deterministic sequence of target states $\{X_{-w}(t), \cdots X_{-w}(t+\Delta_t)\}$ by simulating the mean path, starting from $X_{-w}(t) = E[X(t)|Z(t-1)]$ and assuming noiseless dynamics, i.e., $X_{-w}(t+j+1) = f(X_{-w}(t+j))$. From this state sequence and the sampled configuration state V_i^m , we obtain a deterministic sequence of future measurements $\{z_{-\nu}(t), \cdots z_{-\nu}(t+\Delta_t)\}$ by assuming zero measurement noise. This sequence of measurements is then used to obtain a sequence of posterior PDFs

$P(X(t+j)|Z(t-1), z_{-\nu}(t) : z_{-\nu}(t+j))$ where $j = 0, \cdots, \Delta_t$. An estimate of the sequence of expected posterior information matrices corresponding to the configuration sequence V_l^m can be calculated from these PDFs. The approximate expected information gain $\bar{I}_{\Delta_t}(t, V_i(m))$ is then arrived at as the sum of incremental changes in the determinants of these information matrices during each measurement update. Given $\bar{I}_{\Delta_t}(t, V_i(m))$, p-MCMC computes the value of target density for the configuration $V_i(m)$ at t as

$$\pi_t(V_i(m)) = Ce^{(\frac{\bar{I}_{\Delta_t}(t, V_i(m))}{T})}, \tag{8.11}$$

where C and T are positive constants.

The p-MCMC algorithm runs several such chains, exploring a large volume of the configuration space. As the chains are run simultaneously on parallel processors, there is no additional time penalty involved [7]. After simulating each chain for a sufficiently large number of time steps T_{total}, we assemble the collection of N_{par} Markov chains $V_{total}(t) = \{(V_1^0 \cdots V_1^{T_{total}}), \cdots, (V_{N_{par}}^0 \cdots V_{N_{par}}^{T_{total}})\}$. Then the optimal sensor configuration at time t can be approximated as

$$V^*(t) \approx \arg\max_{V_i(m)} \bar{I}_{\Delta_t}(t, V_i(m)) \quad V_i(m) \in A = V_{total}(t) \tag{8.12}$$

From $V^*(t)$ we obtain the optimal sensor configuration at time t that maximizes the information gain until $t + \Delta t$. The optimal action for the time $t + 1$ is computed in a similar manner by considering the information gain over the time window between $t+1$ and $t+1+\Delta t$. We test the effectiveness of the p-MCMC method in an example sensor scheduling problem. A sensor located on earth is required to track 10 moving targets. The sensor has a field of regard (FOR) of π radians and a field of view (FOV) 15° in each direction, as shown in Figure 8.19. The orbits of the targets are

randomly sampled with a mean semimajor axis of length 42164 km and standard deviation of 100 km. The inclination of the orbits are sampled uniformly from an interval $[-\pi/6, \pi/6]$ radians.The eccentricity and eccentricity anomaly for each orbit is sampled from uniform distributions in the intervals $[0.01, 0.04]$ and $[-\pi/4, \pi/4]$ radians respectively. Each target PDF is initialized with a diagonal covariance matrix constructed using randomly sampled standard deviations. The standard deviations for the position coordinates are sampled uniformly from the range $[10\text{km}, 80\text{km}]$ and for velocity coordinates are sampled from the range $[0.01\text{km/s}, 0.1\text{km/s}]$. The process noise is modeled as acceleration terms with zero mean and variance equal to $10^{-10}\frac{km^2}{s^4}$.

The sensor records measurements during a periodic observation window, with the

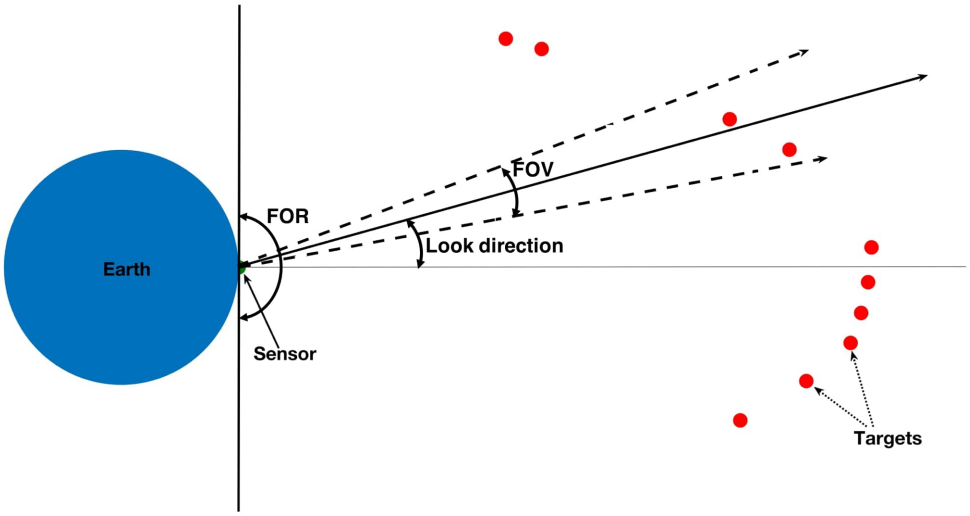


Figure 8.19: Sensor tasking system with 10 targets and a sensor with FOR = π rad and FOV=15° around the look direction. Reprinted with permission from [7].

period of the measurement cycle being 600 s. The observation window of the sensor

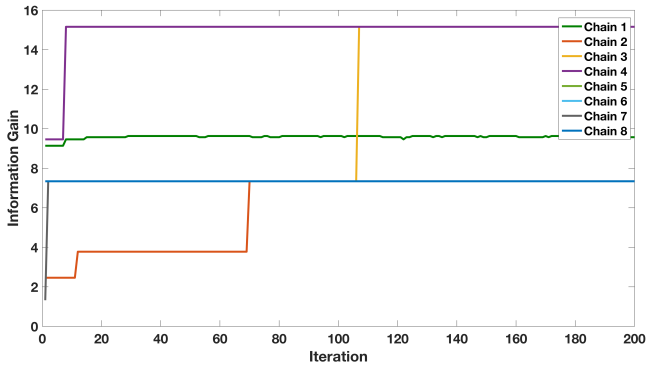


Figure 8.20: Expected information gain at different parallel pool workers. Reprinted with permission from [7].

comprises 3 measurements, spaced 2 s apart. The measurement noise associated with the observations is set at 2.5893×10^{-6}. The observation model is allowed to run for 5 measurement cycles, that is 3000 s. Eight parallel workers were used to run MCMC simulations with different randomly sampled initial conditions. A PGM-I filter was used to track the targets. The simulations were run using the Terra supercomputing cluster maintained by High Performance Computing Resources (HPRC) at Texas A&M University.

The expected information gain over the MCMC runs in the eight parallel pool workers is shown in Figure 8.20. Each Markov chain corresponds to a different initial look direction. All chains are seen to converge to local optima in just a few iterations. Some of the chains are also seen to converge to the same local optimum. For example, chains 2, 5, 6, 7, 8 converge to a local optima with a value of 7 for expected information gain.

The instantaneous net information gain over all ten targets, resulting from the optimal look direction is shown in Figure 8.21. The spikes correspond to the measurement windows that are periodical with time period 600s. The time evolution

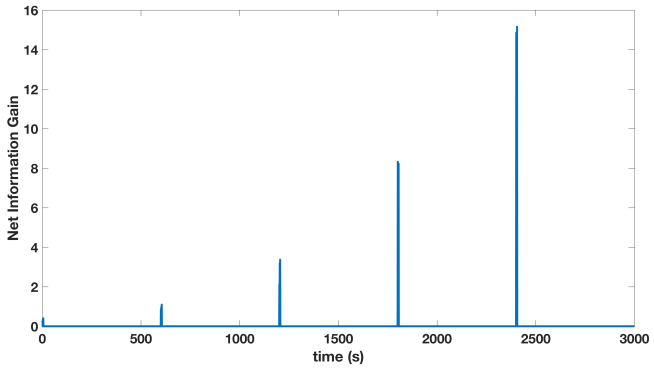


Figure 8.21: Instantaneous net information gain, with spikes corresponding to the five measurement windows. Reprinted with permission from [7].

of the determinant of the information matrix, for each of the ten targets is shown in Figure 8.22. The information gain is measured as the change in determinant of the information matrix during a measurement update. As expected, the optimal look direction does not capture all targets in a particular measurement window. For example, during the first measurement cycle that lasts between 1s and 5s, three targets (Targets 2, 5 and 6) show an increase in information. However, during the measurement window between 601s and 605s, only two targets (Targets 1 and 7) are being observed. The time evolution of the sum of determinants of the realized information gains over all ten targets are shown in Figure 8.23.

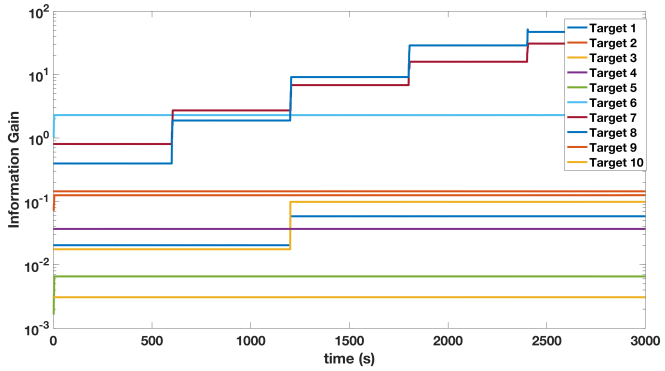


Figure 8.22: Time evolution of the determinant of the information matrix for the ten targets. Reprinted with permission from [7].

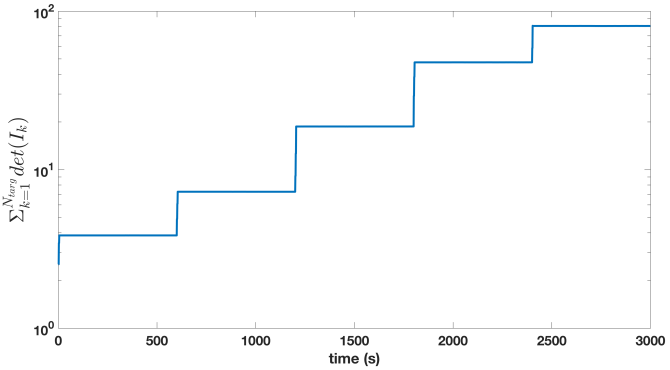


Figure 8.23: Time evolution of determinants of information matrices summed over all targets. Reprinted with permission from [7].

9. CONCLUSIONS

In this chapter, we summarize the contributions of this dissertation and discuss avenues for future research.

In this dissertation we developed a novel Particle Gaussian Mixture (PGM) filtering framework for addressing the problem of nonlinear Bayesian estimation. The PGM filtering design was inspired by our work on the unscented Kalman particle hybrid filter. The hybrid filter, whose development is discussed in chapter 3 was proposed to tackle the problem of space object tracking. It employs the UKF for tracking when the target is inside the FOV of the observer. In order to handle the nonlinear distortion outside the FOV, the tracking scheme transitions from UKF to PF as the object exits the FOV. It addresses the problem of particle depletion through a suitably designed PF to UKF transition scheme based on a Kalman update. The hybrid filtering scheme was employed to estimate the state of a space object in inclined low earth orbits and the estimation performance is studied in terms of the RMSE and NEES metrics. The hybrid filters that employed the Kalman measurement update were found to offer reliable estimation performance even with large initial uncertainty and sparse measurements while using a relatively small number of particles. The superior performance of hybrid filter in comparison to UKF and the PF particle approximation lead us to the following important conclusions.

- Conventionally employed nonlinear filters such as the EKF and UKF are not effective in estimation problems in which the state PDF undergoes extensive distortion under nonlinear transformations.

- Monte Carlo uncertainty propagation can handle the nonlinear transformations without making restrictive assumptions about state PDF.

- In the absence of a good importance density, the performance of particle measurement update deteriorates quickly with the dimension of the state space.
- Kalman measurement update, even though inexact in nonlinear settings, can prevent filter collapse and produce consistent estimates.

Based on these findings, in chapter 4 we designed a generalization of the hybrid filter, namely the PGM-I filter to address the general multimodal nonlinear filtering problem. Like the hybrid filter, the PGM-I 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-I approach allows the number and weight of mixture components to be adapted during propagation unlike the conventional mixture filters [18, 19]. Additionally, the PGM-I filter is not prone to the curse of dimensionality associated with particle measurement updates. The PGM-I 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-I filter is employed in three test cases to evaluate the estimation performance. It is demonstrated that the PGM-I filter offers superior estimation performance in comparison to UKF, PF the blob filter and a mixture UKF. The PGM-I filter is demonstrated to be capable of tracking the 40 dimensional Lorenz 96 system wherein the PF and blob filter suffers weight depletion.

The PGM-I filter assumes the validity of the Kalman measurement update even when measurements are highly nonlinear. The can prove restrictive in the presence of significant measurement nonlinearity. In chapter 5 we proposed the PGM-II filter which relaxes this assumption. Like its predecessor, the PGM-II filter uses the transition kernels of the underlying Markov chain to generate samples during the

propagation step. The samples are then clustered to recover a GMM representation of the propagated prior PDF. The measurement update is performed with the help of a parallel MCMC based sampling algorithm. As a result, the PGM-II measurement update step is asymptotically exact. It does not enforce restrictive assumptions on the number of mixture components during either propagation or measurement update. Using the exponential forgetting assumption, we proved the convergence of the PGM-II filtered PDF to the true filter PDF. The PGM-II filter is employed in the estimation of two test cases to evaluate the estimation performance. The PGM-II filter is seen to outperform the PF and the UKF in both test cases. The blob filter is seen to offer superior performance in the blind tricyclist problem. Our results from this chapter also indicated that strategies for improving the performance of the MCMC method in sampling extremely multimodal target densities need to be studied as future work.

In chapter 6 we extensively studied the performance of the PGM filters on a selected class of benchmark problems chosen from recent literature. We compared the PGM filtering performance with that of other recently proposed general purpose nonlinear filters such as the log homotopy based particle flow filters and the feedback particle filter. The performance of PGM filters was found to be at par with or better than these despite being much simpler in implementation. The benchmark study allowed us to investigate how dimensionality, nonlinearity and number of samples affect the performance of PGM filters. Based on our results, we found the following important guidelines for choosing between the PGM filters.

- For problems with significant nonlinearity in propagation but mildly nonlinear measurement functions, choose PGM-I filter.

- When the measurement nonlinearity is significant or when measurement noise

is significantly multimodal, choose PGM-II filter.

- However, when the computational cost is a concern such as when estimation is to be performed in very high dimensions, PGM-I filter may be chosen over PGM-II.

- A non-Gaussian measurement noise can be incorporated within the existing PGM-I framework. To accomplish this, the noise has to be approximated by a Gaussian via moment matching. However, the PGM-I performance in this case can turn out to be inferior in comparison to the PGM-II filter.

We found that filters that rely on Kalman measurement update may provide more consistent estimates than those that use sampling. This is despite the latter being the asymptotically exact choice. Note that the PGM-I filter is designed assuming that the measurement noise is Gaussian. It may be possible to represent a non-Gaussian measurement noise with a Gaussian mixture model. A rigorous generalization of the PGM-I filter that can handle Gaussian mixture measurement noise may be pursued as future work. The results obtained from these experiments inspired us to design an augmented PGM-I filter, a variant of PGM-I that can better handle measurement nonlineariy/non-Gaussianity without incurring the significantly higher computational cost of PGM-II filter. The augmented PGM-I filter uses samples from the conditional measurement distribution to obtain samples from the joint distribution of state and measurement random variables. It then uses density estimation to obtain a mixture representation of the joint distribution from which the posterior PDF is obtained. The augmented PGM-I filter only assumes the availability of a density estimation algorithm that can accurately recover the joint PDF. The augmented PGM-I can circumvent the need for the aforementioned PGM-I generalization thgat incorporate multimodal measurement noise.

Given the central role the it plays in PGM filtering, we investigated how certain aspects of the density estimation problem influence the PGM filtering performance. The maximum number of mixture modes that can be used to represent the state PDF is the main user defined input in the density estimation step. We investigated how this parameter affects the PGM filtering performance. We found that for one dimensional problems, increasing the number of modes improved the filtering performance of PGM-I filters. This is found to be true even when the actual state PDF is unimodal. This improvement in PGM-I performance can be ascribed to the splitting of measurement function domain that happens when the PDF is represented using more modes. When the domain is split, the linearization errors become smaller and the posterior PDF becomes more accurate. The effect is pronounced in augmented PGM-I as the measurement nonlinearities are incorporated in the clustering distribution resulting in more effective splitting of the domain. However, a similar effect was not observed in PGM-II as the MCMC based measurement update is not subject to linearization errors. Our results from this experiment indicate the possibility of using an augmented PGM-I filter with a relatively large number of modes as a substitute for PGM-II to save computational cost. We experimented further by designing PGM filters that rely on non-parametric density estimation methods. In particular we considered a a PGM-II variant that uses kernel density estimation and MCMC to perform measurement update. The KDE performance was found to be remarkably sensitive towards the kernel window size selection. However, this experiment underlined the flexibility of PGM-II design with respect to the choice of density estimators.

Next we considered developing a PGM-I analogue for the distributed filtering problem. In particular, we explored the decentralized estimation scenario, in which the network of agents/sensors is assumed to have no central node. We derived the

equations for performing distributed averaging of information vectors and matrices when the prior PDF is multimodal and the measurement function is nonlinear. We considered a simple example where a one step measurement update is performed using these equations. Future research must address the problem of developing distributed PGM-I, PGM-II filters that allows connectivity failures and distributed clustering.

In chapter 8, we looked at the parallel MCMC method that we developed for PGM-II filtering. Even though MCMC is an inherently sequential algorithm, the increasing prevalence of parallel computing has bolstered efforts towards parallelizing it. We demonstrated how the parallel MCMC method is significantly more successful in sampling from multimodal densities in comparison to a single Metropolis-Hastings based random walk chain. We also explored how the parallel MCMC approach can be utilized to solve global optimization problems. The performance of the parallel MCMC method was tested on multiple optimization examples and compared with that of simulated annealing. Strict theoretical guarantees on the finite time sampling performance of the parallel MCMC method have to be pursued as future work.

REFERENCES

[1] D. Raihan and S. Chakravorty, "Particle Gaussian mixture filters-I," *Automatica*, vol. 98, pp. 331–340, 2018.

[2] D. Raihan and S. Chakravorty, "Particle Gaussian mixture (pgm) filters," in *Proceedings of the 2016 19th International Conference on Information Fusion (FUSION)*, pp. 1369–1376, 2016.

[3] D. Raihan and S. Chakravorty, "Particle Gaussian mixture filters-II," *Automatica*, vol. 98, pp. 341–349, 2018.

[4] D. Raihan and S. Chakravorty, "Particle Gaussian mixture filters-II," in *Proceedings of the 2018 21st International Conference on Information Fusion (FUSION)*, pp. 1092–1099, 2018.

[5] D. R. A. Veettil and S. Chakravorty, "A ukf-pf based hybrid estimation scheme for space object tracking," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, pp. 4221–4240, August 2015.

[6] A. V. D. Raihan and S. Chakravorty, "An unscented Kalman-particle hybrid filter for space object tracking," *The Journal of the Astronautical Sciences*, vol. 65, pp. 111–134, 2018. url: https://doi.org/10.1007/s40295-017-0114-8.

[7] D. Raihan, W. Faber, S. Chakravorty, and I. Hussein, "Parallel Markov chain Monte Carlo for sensor scheduling." Presented at the *AAS/AIAA Astrodynamics Specialist Conference*, August 2018.

[8] T. Lefebvre, H. Bruyninckx, and J. D. Schutter, "Comment on "a new method for the nonlinear transformation of means and covariances in filters and esti-

mators"," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406 – 1409, 2002.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[10] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Transactions of ASME-Journal of Basic Engineering*, vol. 83, pp. 96–108, 1961.

[11] G. Smith, S. Schmidt, and L. McGee, "Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle," Tech. Rep. NASA TR-135, NASA, Jan. 1962.

[12] S. J. Julier, J. K. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference*, pp. 1628–1632, 1995.

[13] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004.

[14] E. Wan and R. V. D. Merwe, "The unscented Kalman filter," in *Kalman Filter and Neural Networks* (S. Haykin, ed.), pp. 221–280, New York: J. Wiley and Sons, 2001.

[15] M. L. Psiaki, "The blind tricyclist problem and a comparative study of nonlinear filters," *IEEE Control Systems Magazine*, vol. 33, pp. 40–54, June 2013.

[16] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254 –1269, 2009.

[17] J. Dunik, O. Straka, M. Simandl, and E. Blasch, "Random-point-based filters: analysis and comparison in target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1403 –1421, 2015.

[18] H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[19] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[20] G. Terejanu, P. Singla, T. Singh, and P. Scott, "Adaptive Gaussian sum filter for nonlinear Bayesian estimation," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2151–2156, 2011.

[21] K. DeMars, R. Bishop, and M. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance Control and Dynamics*, vol. 36, pp. 1047–1056, 2013.

[22] M. L. Psiaki, "Gaussian mixture nonlinear filtering with resampling for mixand narrowing," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 1047–1056, 2016.

[23] N. Gordon, D. Salmond, and A. Smith, "A novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proceedings F, Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[24] 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*, vol. 50, no. 2, pp. 174–188, 2001.

[25] 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*, vol. 2, pp. 316–334, 2008.

[26] G. Evensen, *Data Assimilation:The Ensemble Kalman Filter*. Berlin: Springer, 2002.

[27] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2465 –2480, 2013.

[28] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[29] H. Schaub and J. Junkins, *Analytical Mechanics of Space systems*. Reston VA: AIAA, 2003.

[30] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10143–10162, 1994.

[31] G. Burgers, P. V. Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly Weather Review*, vol. 126, no. 6, pp. 1719–1724, 1998.

[32] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3562–3568, 2006.

[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley-Interscience, 2 ed., November 2000.

[34] A. K.Jain, M. N. Murthy, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 13, no. 3, pp. 264–323, 1999.

[35] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[36] O. Kallenberg, *Foundations of Modern Probability*. New York: Springer, 1997.

[37] R. Vershynin, *High Dimensional Probability : An Introduction with Applications in Data Science*. Cambridge, UK: Cambridge University Press, 2017.

[38] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high- dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.

[39] P. Bickel, B. Li, and T. 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*, vol. 3, pp. 318–329, 2008.

[40] J. L. Anderson, "Ensemble Kalman filters for large geophysical applications," *IEEE Control Systems Magazine*, vol. 29, pp. 66–82, 2009.

[41] D. Pelleg and A. 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*, pp. 277–281, 1999.

[42] U. D. Hanebeck, K. Briechle, and A. Rauh, "Progressive Bayes: a new framework for nonlinear state estimation," in *SPIE vol.5099 Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, pp. 256–267, 2003.

[43] E. N. Lorenz, "Predictability: a problem partly solved," in *Proceedings of Seminar on Predictability*, vol. 1, pp. 1–18, 1996.

[44] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall, 1996.

[45] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[46] W. K. Hastings, "Monte Carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[47] D. N. VanDerwerken and S. C. Schmidler, "Parallel Markov chain Monte Carlo," *ArXiv e-prints*, Dec. 2013. preprint, url: https://arxiv.org/abs/1312.7479.

[48] J. G. Propp and D. B. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Structures & Algorithms*, vol. 9, pp. 223–252, 1996.

[49] A. V. D. Raihan and S. Chakravorty, "Particle Gaussian mixture (pgm) filters," *ArXiv e-prints*, 2016. preprint, url: http://arxiv.org/abs/1603.04510.

[50] R. V. D. Merwe, A. Doucet, N. D. Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in Neural Information Processing Systems*, pp. 584–590, 2001.

[51] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proceedings of SPIE Signal and Data Processing of Small Targets*, September 2007.

[52] A. Gelfand and D. Dey, "Bayesian model choice: asymptotics and exact calculations," *Journal of the Royal Statistical Society. Series B*, vol. 56, pp. 501–514, 1994.

[53] M. Khan and M. Ulmke, "Non-linear and non-Gaussian state estimation using log-homotopy based particle flow filters," in *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, 2014.

[54] S. Pal and M. Coates, "Particle flow particle filter for Gaussian mixture noise models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp. 4249–4253, 2018.

[55] K. Berntorp, "Feedback particle filter: application and evaluation," in *Proceedings of 2015 18th International Conference on Information Fusion*, pp. 1633–1640, 2015.

[56] F. E. D. Melo, S. Maskell, M. Fasiolo, and F. Daum, "Stochastic particle flow for nonlinear high-dimensional filtering problems," *ArXiv e-prints*, Nov. 2015. preprint, url: https://arxiv.org/abs/1511.01448v3.

[57] A. Tamjidi, R. Oftadeh, S. Chakravorty, and D. Shell, "Efficient distributed state estimation of hidden Markov models over unreliable networks," in *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pp. 112–119, 2017.

[58] L. Xiao, S. Boyd, and S. Lall., "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of 2005 Fourth International Symposium on Information Processing in Sensor Networks*, pp. 63–70, 2005.

[59] G. W. Basse, N. S. Pillai, and A. Smith, "Parallel Markov chain Monte Carlo via spectral clustering," in *Artificial Intelligence and Statistics*, pp. 1318–1327, 2016.

[60] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[61] S. Surjanovic and D. Bingham, "Virtual library of simulation experiments: test functions and datasets." Retrieved January 9, 2019, from `http://www.sfu.ca/~ssurjano`.

[62] M. Vali, "Sub-diving labeling method for optimization problem by genetic algorithm," *ArXiv e-prints*, 2013. preprint, url: https://arxiv.org/abs/1307.5840.

[63] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test functions for optimization needs*, vol. 101, 2005.

[64] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

APPENDIX A

CLUSTERING

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

$$S = \{x_1, \cdots, x_i, \cdots, x_n\}, \quad x_i \in \mathbb{R}^{d\times 1}. \tag{A.1}$$

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 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.

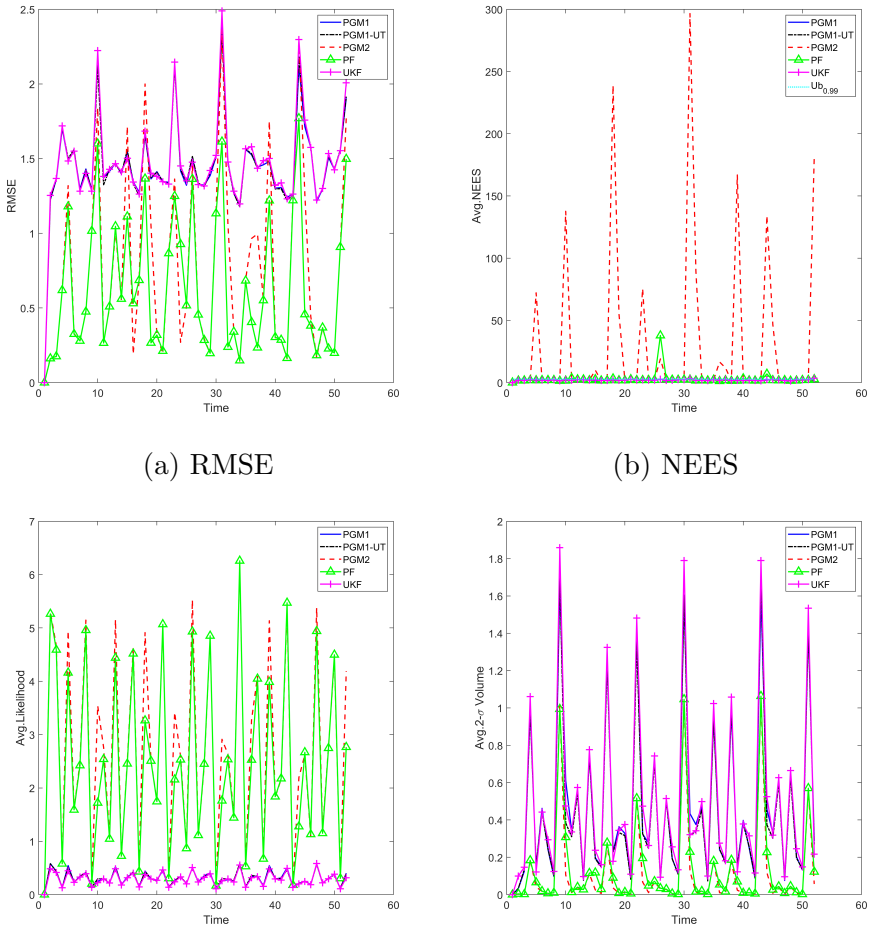
The Expectation-Maximization (E-M) algorithm is an iterative approach to arriving at a GMM describing a data set using probabilistic methods [33]. The E-M algorithm computes the modal parameters Θ that maximizes the log likelihood of the observed data S. As the name suggests, each iteration of the EM 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 alogrithms 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 [64]. 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, choice of clustering algorithms has to be exercised based on the nature of application at hand, expected accuracy and available computational resources.

APPENDIX B

SIMULATION RESULTS

B.1 Generalized bi-modal system model



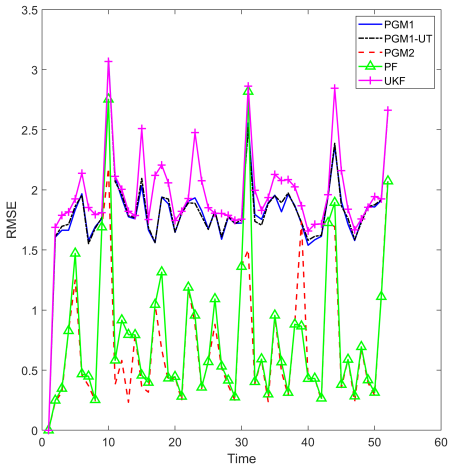
(a) RMSE

(b) NEES

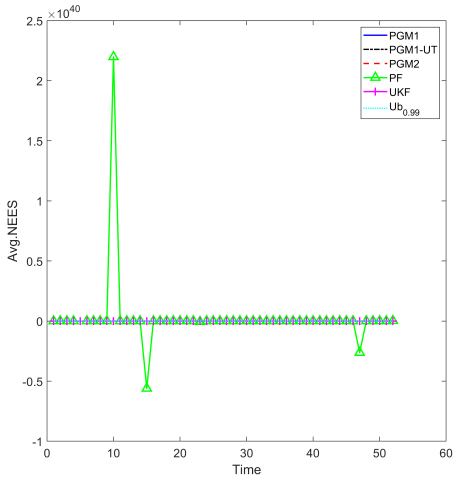
(c) Likelihood

(d) $2-\sigma$ volume

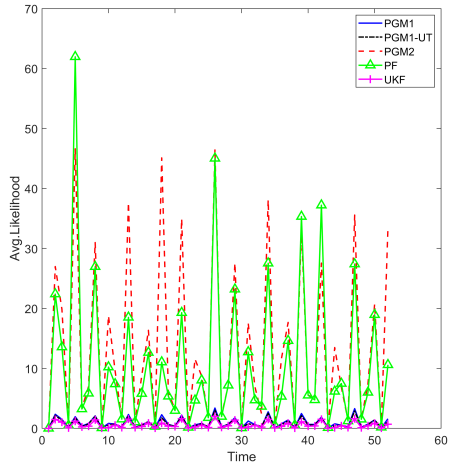
Figure B.1: Monte Carlo averaged performance metrics for the bi-modal system model at d=2



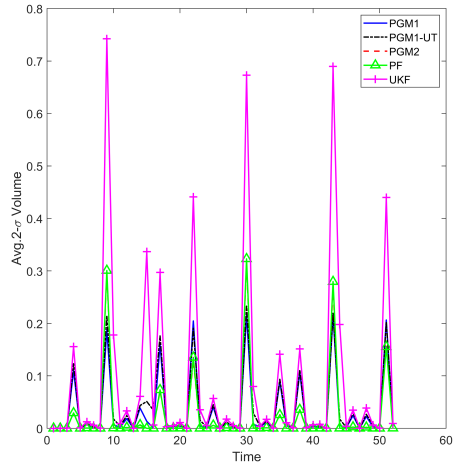
(a) RMSE



(b) NEES

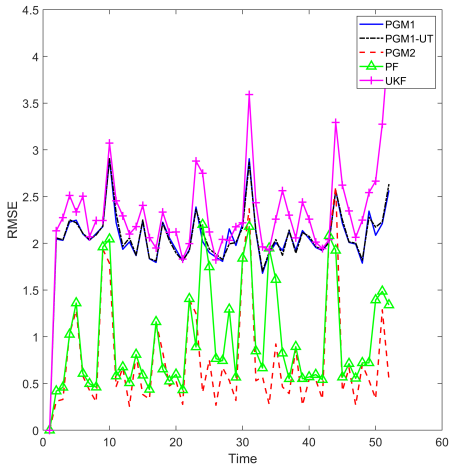


(c) Likelihood

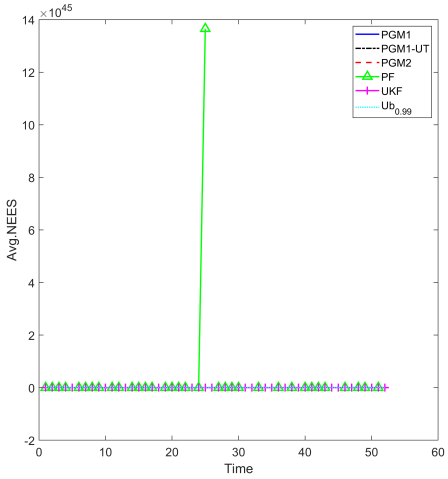


(d) $2-\sigma$ volume

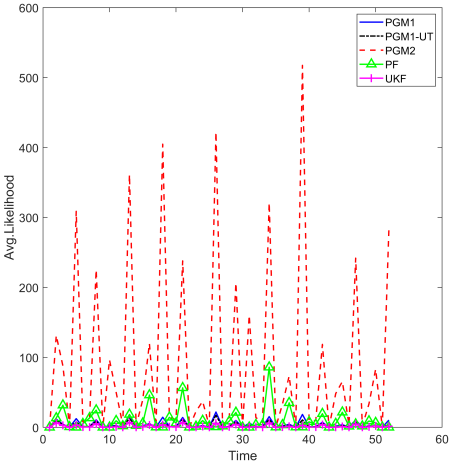
Figure B.2: Monte Carlo averaged performance metrics for the bi-modal system model at d=4



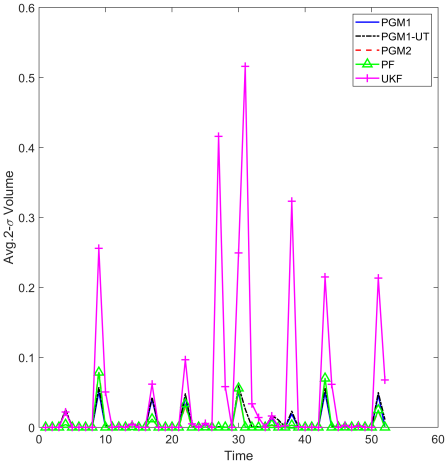
(a) RMSE



(b) NEES

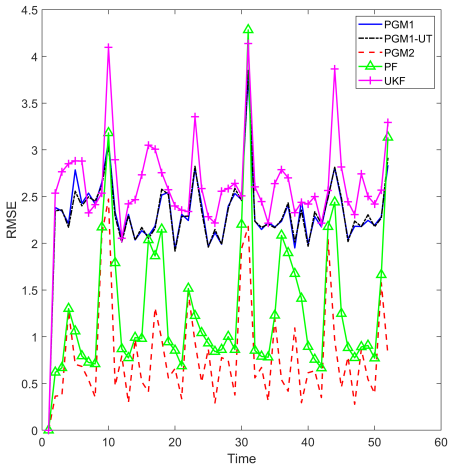


(c) Likelihood

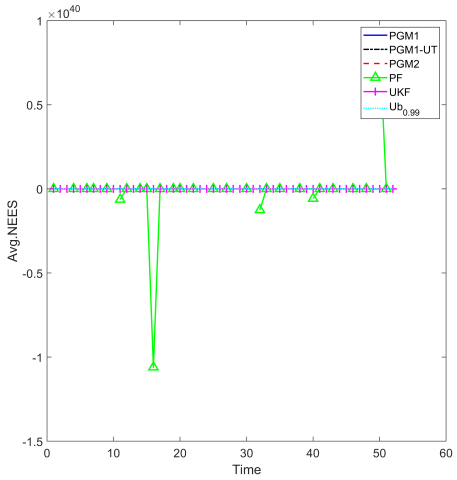


(d) $2-\sigma$ volume

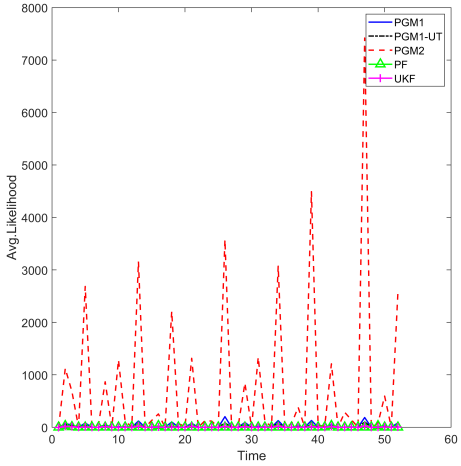
Figure B.3: Monte Carlo averaged performance metrics for the bi-modal system model at d=6



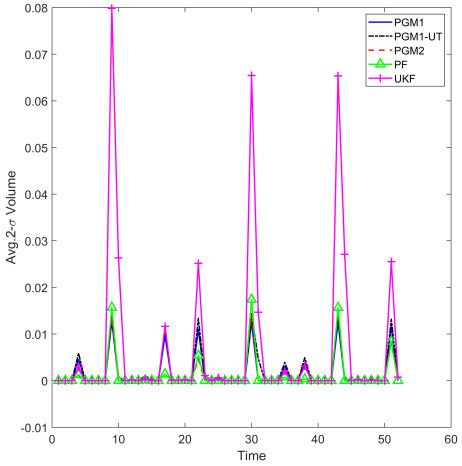
(a) RMSE



(b) NEES

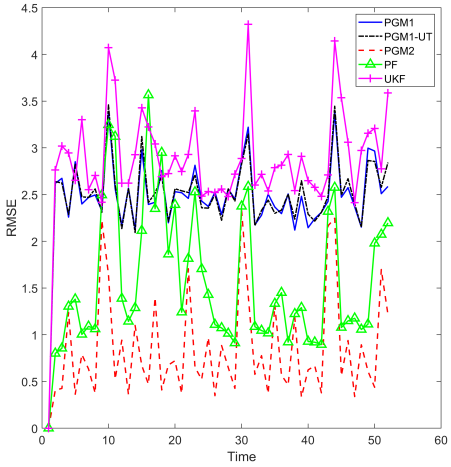


(c) Likelihood

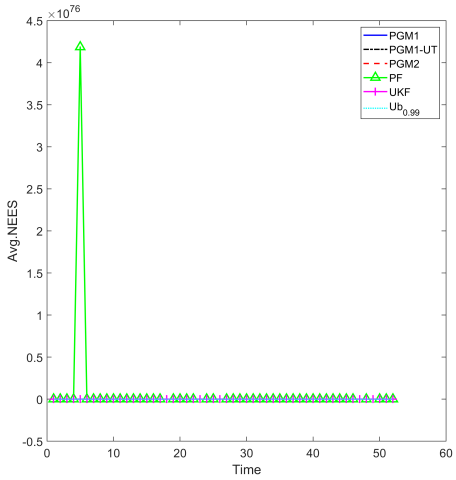


(d) $2-\sigma$ volume

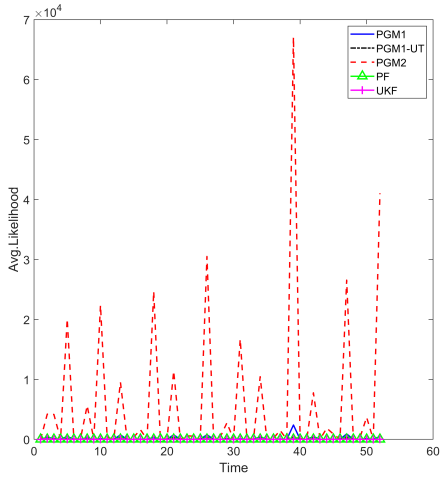
Figure B.4: Monte Carlo averaged performance metrics for the bi-modal system model at d=8



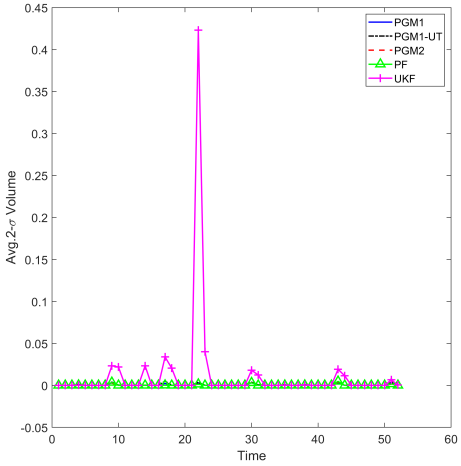
(a) RMSE



(b) NEES

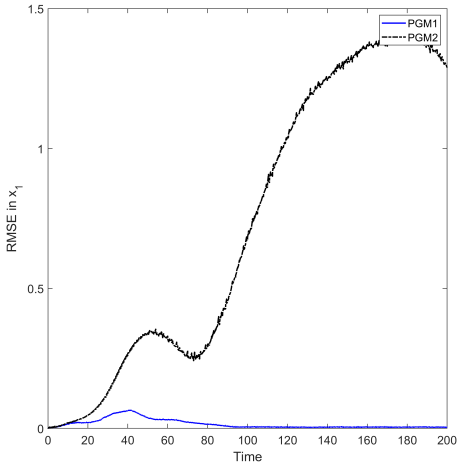


(c) Likelihood

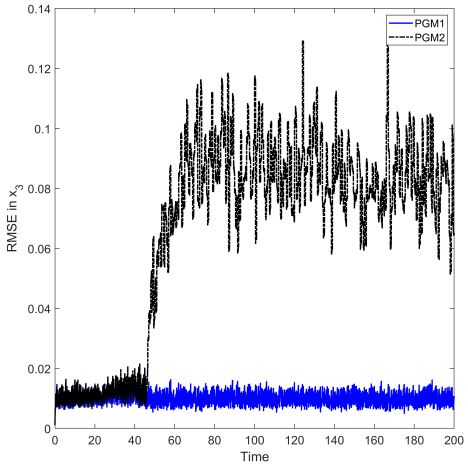


(d) $2-\sigma$ volume

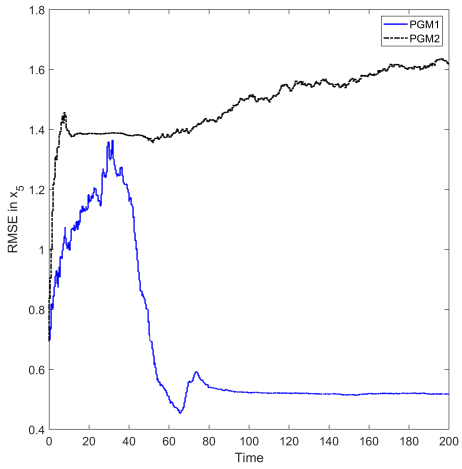
Figure B.5: Monte Carlo averaged performance metrics for the bi-modal system model at d=10



(a) x_1

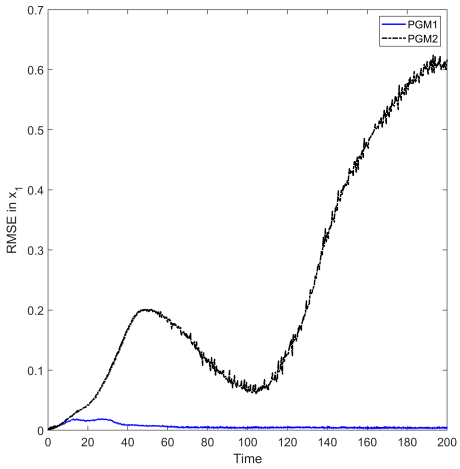


(b) x_3

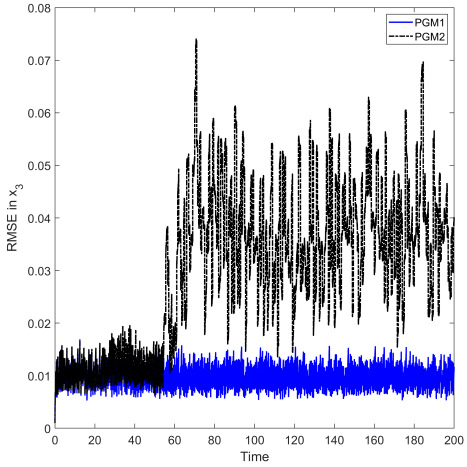


(c) x_5

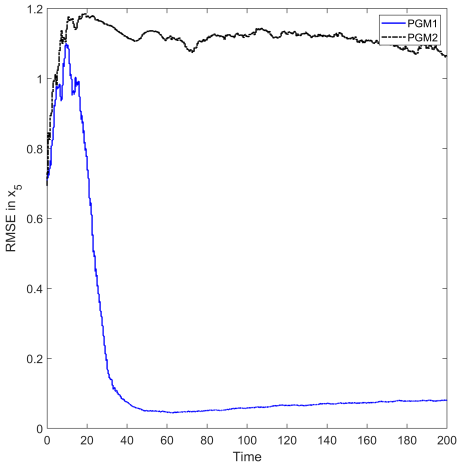
Figure B.6: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 50$



(a) x_1

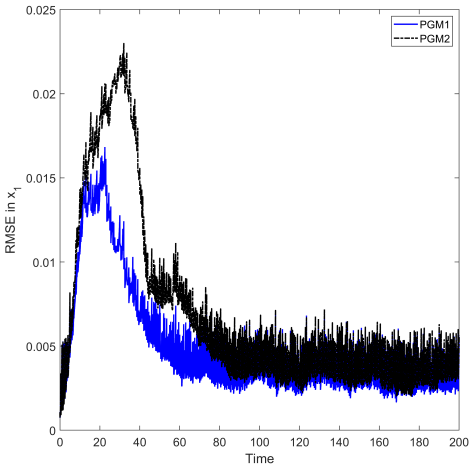


(b) x_3

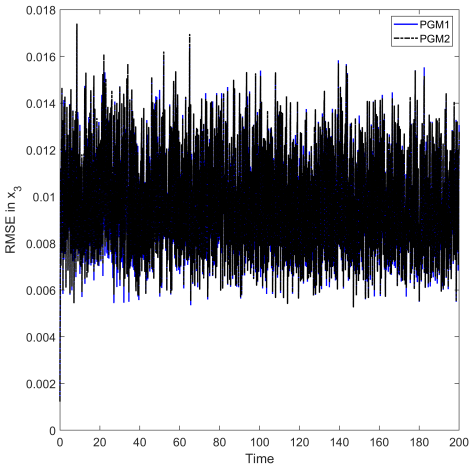


(c) x_5

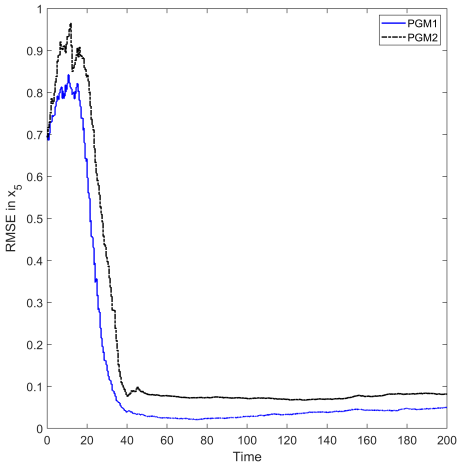
Figure B.7: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 100$



(a) x_1

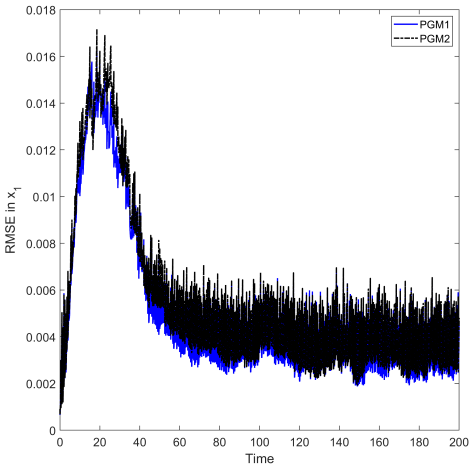


(b) x_3

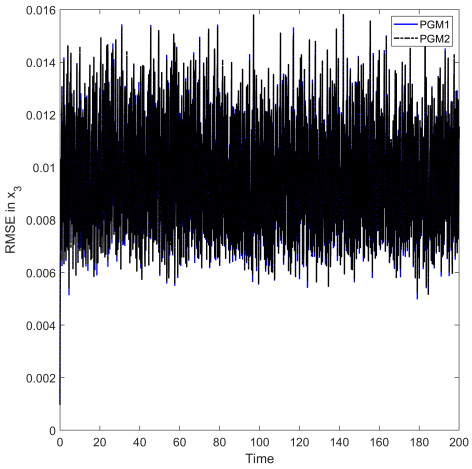


(c) x_5

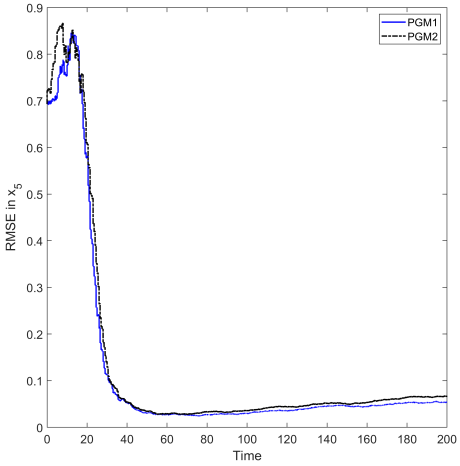
Figure B.8: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 500$



(a) x_1



(b) x_3



(c) x_5

Figure B.9: Monte Carlo averaged RMSE in x_1, x_3 and x_5 when $N_p = 1000$