View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of New Orleans

University of New Orleans

ScholarWorks@UNO

University of New Orleans Theses and Dissertations

Dissertations and Theses

8-5-2010

Application of Dirichlet Distribution for Polytopic Model Estimation

Jaipal Katkuri

University of New Orleans

Follow this and additional works at: https://scholarworks.uno.edu/td

Recommended Citation

Katkuri, Jaipal, "Application of Dirichlet Distribution for Polytopic Model Estimation" (2010). *University of New Orleans Theses and Dissertations*. 1210.
https://scholarworks.uno.edu/td/1210

This Thesis is protected by copyright and/or related rights. It has been brought to you by ScholarWorks@UNO with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in University of New Orleans Theses and Dissertations by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.

Application of Dirichlet Distribution
for Polytopic Model Estimation

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Engineering

by

Jaipal R. Katkuri

B.E. Osmania University, 2006

August, 2010

Copyright 2010, Katkuri

Dedication

To my parents, my sister, Chinna, and all the teachers for their support, encouragement and the corvette they're giving me for graduation.

"*The ideal situation occurs when the things that we regard as beautiful are also regarded by other people as useful*" – Donald Knuth

Acknowledgments

It is indeed a great pleasure to thank all those who have, directly or indirectly, helped me in successfully completing thesis.

At the outset I wish to thank my advisor Dr. V. P. Jilkov who was by my side always, patiently and constantly inspiring, encouraging and guiding me throughout my Master's program. I have learnt a lot from his meticulous planning and implementation, dedication and hard work. My association with him for over two years was a rewarding experience. Special thanks are extended to Dr. X. R. Li for his technical advice throughout my Master's research work and also for the constructive and valuable comments on the thesis. I also thank Dr. H. Chen and Dr. Charalampidis and members of Information and System Laboratory (ISL) for their guidance and help.

I would like to acknowledge and thank all of my friends, especially Dr. Chintu, Ani, Murpal, Naveen, Aashay, Gowti, Monty, Vandu, and last but not least Appu for their constant encouragement and moral support and for being with me through ups and downs.

Last, and most important of all, I wish to thank my parents, my sister and Chinna without those co-operation this thesis would have been highly impossible.

Contents

Abstract

The polytopic model (PM) structure is often used in the areas of automatic control and fault detection and isolation (FDI). It is an alternative to the multiple model approach which explicitly allows for interpolation among local models.

This thesis proposes a novel approach to PM estimation by modeling the set of PM weights as a random vector with Dirichlet Distribution (DD). A new approximate (adaptive) PM estimator, referred to as a Quasi-Bayesian Adaptive Kalman Filter (QBAKF) is derived and implemented. The model weights and state estimation in the QBAKF is performed adaptively by a simple QB weights' estimator and a single KF on the PM with the estimated weights. Since PM estimation problem is nonlinear and non-Gaussian, a DD marginalized particle filter (DDMPF) is also developed and implemented similar to MPF. The simulation results show that the newly proposed algorithms have better estimation accuracy, design simplicity, and computational requirements for PM estimation.

KEY WORDS: Model interpolation, Quasi-Bayes procedure for mixtures, Dirichlet distribution, Jump-Markov linear systems, Polytopic model

List of Figures

Chapter 1

Introduction

Research on multiple-model (MM) approach has spread out in many areas in last decades. The reason is in the elegant solutions that MM approach provides for modeling, estimation, and control. The MM framework can give appealing solutions to almost all filtering problems. A canonical application of the MM approach is target tracking. An elaborate, in-depth, explanation of different state-of-the-art MM algorithms and their application to target tracking problems can be found in [1]. Another important application of the MM framework is system fault detection and isolation (FDI), e.g., [2], [3].

Most existing MM estimation algorithms provide a solution to the problem of estimating the state and the mode of a hybrid, usually Markov jump system (MJS), [4]. A possible alternative is to use another model structure which relies on representing (approximating) a possible truth through explicit interpolation between models. For example, if the weighed combinations of the local models in the MJS model set correspond to physically relevant conditions, which are not accounted for in the MJS model set, it is desirable to interpolate among these models. Such a structure is, sometimes, referred to as a blended MM structure [5]. In this structure, the model that is valid is a weighed combination of some local models in a chosen (designed) model set [6], [7]. When the combinations are restricted to be convex, a subset of blended MM structure is created that is usually named as Polytopic Model (PM) or Convex Model (CM). The convexity restriction is added to ensure that the state estimates based on the ones of the local models are fully valid and the rest (non-convex combinations) are not. Polytopic models have been most often used in the area of FDI [8], [9], [10], [11], [12], [13], [14], [15]. The main motivation for using the PM framework in FDI is that it allows for a large class of fault conditions to be modeled by using a relatively small numbers of local models, and no explicit modeling of system mode evolution (e.g., Markov chain model transitions) is required. Estimation using PM structure consists of estimating both the state and model weights of the local models directly. This kind of estimation problem is nonlinear and non-convex due to the products of the state and model weights. There exist a number of methods for PM estimation in the literature. An overview of such methods and numerous issues associated with their use is provided in Chapter 2.

In this thesis, we propose a novel approach to PM estimation. The main idea is to model the set of PM weights as a random vector with Dirichlet prior probability distribution. The Dirichlet distribution (DD) [16] is the most natural model for probability weights of mixtures in Bayesian statistics [17]. It inherently satisfies the requirements of a probability mass function and offers a great variety of shapes and flexibility to model different situations (See Chapter 2). Two estimation methods are carried out.

Under the assumption that the PM weights obey a DD, a new approximate (adaptive) PM estimator is derived which is referred to as a Quasi-Bayesian Adaptive Kalman Filter (QBAKF). The operation of the QBAKF filter is based on two main techniques: quasi-Bayesian estimation of mixture probabilities [18] and approximating the vertex-model likelihoods through a single Kalman filter (KF). The model weights and state estimation in the QBAKF filter is performed adaptively by a simple QB weights' estimator and a single KF on the PM with the estimated weights. Since only one KF used, the QBAKF filter is computationally very efficient even for large vertex-model sets – it is almost independent of the model set size.

Since PM estimation problem is nonlinear and non-Gaussian, a particle filter (PF) is a viable option to apply. A generic PF is not a good choice because for large number of models the dimension of the particles (state dimension + number of models) becomes large and possibly a prohibitively large number of particles will be needed. However, based on the observation that the PM is conditionally-linear (given the model weights), the Rao-Blackwellization technique [19] can be directly applied. The result is a DD marginalized PF (DDMPF), similar to the one of [20], that performs linear KF for the linear part, given (an estimate of) the model weights; and a PF for the nonlinear part with respect to the model weights only, given (an estimate of) the state. The DDMPF leads to dramatic computation saving as compared to a generic PF. Based again on the DD weights' model, a DDMPF is also obtained and implemented in this thesis.

The effectiveness and improved interpolation properties of the new approach are demonstrated by two examples target tracking. Monte Carlo simulation results are obtained and compared with the well known Autonomous Multiple Model (AMM) [21], Interacting Multiple Model (IMM) [22], and augmented Extended Kalman filters [23].

The remaining part of the thesis is organized as follows. Chapter 2 provides background information on Polytopic models, the Dirichlet distribution, and a generic Quasi-Bayes procedure for mixture probability estimation. Chapter 3 presents the algorithm development of the new QBAKF and the DD marginalized particle filter (DDMPF). Chapter 4 presents Monte Carlo simulation results of two target tracking examples. Finally, Chapter 5 provides summary and conclusions.

Chapter 2

Background

2.1 Polytopic Model (PM)

2.1.1 PM Definition

A commonly used for MM estimation algorithms is the following jump Markov linear system (JMLS) [1]

$$x_{k+1} = F_k^{(i)} x_k + G_k^{(i)} u_k + \Gamma_k^{(i)} w_k^{(i)} \tag{2.1}$$

$$z_k = H_k^{(i)} x_k + I_k^{(i)} u_k + v_k^{(i)} \tag{2.2}$$

where $x_k \in \mathbb{R}^{n_x}$ is the base state, $z_k \in \mathbb{R}^{n_z}$ is the measurement, $u_k^{(i)}$ is a (known) input, $w_k^{(i)}$ and $v_k^{(i)}$ are independent process and measurement noises with $w_k^{(i)} \sim \mathcal{N}(\bar{w}_k^{(i)}, Q_k^{(i)})$, $v_k^{(i)} \sim \mathcal{N}(\bar{v}_k^{(i)}, R_k^{(i)})$.[1] Superscript (i) denotes quantities pertinent to model $m^{(i)}$ in the model set $\mathbb{M} = \{m^{(1)}, m^{(2)}, ..., m^{(r)}\}$, and the jumps, if any, of the system mode have the following transition

[1]Throughout the thesis, $\mathcal{N}(y; \bar{y}, P_y) \triangleq \frac{1}{|2\pi P_y|^{1/2}} \exp[-\frac{1}{2}(y-\bar{y})' P_y^{-1}(y-\bar{y})]$ denotes the pdf of multivariate normal distribution with mean $\bar{y}$ and covariance P_y.

probabilities

$$P\{m_{k+1} = m^{(j)} | m_k = m^{(i)}\} = \pi_{ij}, \forall m^{(i)}, m^{(j)}, k \tag{2.3}$$

An alternative to the JMLS defined in (2.1)-(2.3) is the following polytopic model (PM), given by [24]

$$x_{k+1} = \sum_{i=1}^{r} \mu_k^{(i)} [F^{(i)} x_k + G^{(i)} u_k] + \Gamma_k w_k \tag{2.4}$$

$$z_k = \sum_{i=1}^{r} \mu_k^{(i)} [H^{(i)} x_k + I^{(i)} u_k] + v_k \tag{2.5}$$

with

$$\mu_k^{(i)} \geq 0, \sum_{i=1}^{r} \mu_k^{(i)} = 1 \tag{2.6}$$

where the model probabilities $\mu_k^{(i)}$ are unknown. The system (2.4)–(2.5) can be written in a more compact form as

$$x_{k+1} = \boldsymbol{\mu}'_k (\mathbf{F}_k x_k + \mathbf{G}_k u_k) + \Gamma w_k \tag{2.7}$$

$$z_k = \boldsymbol{\mu}'_k (\mathbf{H}_k x_k + \mathbf{I}_k u_k) + v_k \tag{2.8}$$

where

$$\boldsymbol{\mu}_k = \left[\mu_k^{(1)}, \mu_k^{(2)}, ..., \mu_k^{(r)}\right]', \quad \mathbf{F}_k = \left[F_k^{(1)}, F_k^{(2)}, ..., F_k^{(r)}\right]', \quad \mathbf{G}_k = \left[G_k^{(1)}, G_k^{(2)}, ..., G_k^{(r)}\right]'$$
$$\mathbf{H}_k = \left[H_k^{(1)} \quad H_k^{(2)}, ..., H_k^{(r)}\right]', \quad \mathbf{I}_k = \left[I_k^{(1)}, I_k^{(2)}, ..., I_k^{(r)}\right]'.$$

Remarks:

The PM structure explicitly uses weighed combinations of the local models contrary to the JMLS structure. If the constraints of the model (2.6) are removed, the PM structure can possibly handle larger class of problems. The PM structure relies on the interpolation properties of a weighted combination to compensate when physically relevant conditions are not explicitly represented in the model set of JMLS. This could be the case, for example, for modeling partial faults in FDI problems.

The PM structure does not use transition probability matrix. This matrix provides the transition probabilities between models in the model set. In theory it is known, but practically unknown and it is a design parameter. Thus, PM requires less prior information as compared to JMLS.

2.1.2 Overview of Approaches for PM Estimation

One straightforward approach is to use state augmentation for joint state and parameter estimation [25], i.e., consider an augmented state-parameter vector $[x', \boldsymbol{\mu}']'$ and use extended Kalman filter (EKF) for the PM (2.7)–(2.8). EKF is done by using linearization as an approximation to the nonlinear problem. At each time instant, the nonlinear augmented state-space model is linearized around the current estimate. In the augmented state, the weight are assumed to evolve as a random walk process. In this approach, assuming weight vectors follow random walk process means they are almost constant with small variation [15] which in general does not happen in reality. The major limitation of this approach is that the EKF produced weight estimates do not satisfy the convexity constraint (2.6). Implementing EKF with inequality constraints is a complicated problem. Usually, a projection onto the set constraints is used which may introduce a significant bias in the weight estimates.

Another approach, also proposed in [15], is based on formulating the joint estimation problem as a constrained maximum a posteriori probability (MAP) estimation. The resulting algorithm, referred to as a dual CMF, assumes lack of prior information available about the model weights. This filter, the dual CMF, uses two linear filtering problems to solve optimization problem.The state and weights are estimated separately in two steps. In each step one parameter is fixed and the other is estimated and vice versa. In both steps, to estimate one parameter the other one is assumed as constant. For the PM constraints (2.6), additional theory for implementing equality and inequality constraints in KF's is used. This approach requires solving multidimensional constraint optimization problem which involves a lot of computation. For the constraints extra care need to be taken, which is itself a challenging problem.

Some methods based on robust observer design via pole assignment can be found in [10], [11], [12], [13], [14]. For example, in [11] a polytopic observer stability is guaranteed by pole assignment established through the Linear Matrix Inequality (LMI) method. These methods are not optimal, in general.

Another approach for adaptive estimation of time varying parameters of linear stochastic dynamic systems was developed in [8]. This approach with PM structure uses single KF wherein hypothesized parameters are updated at each time stage by generating the probability of each hypothesis conditioned on residual process and a given probability of transition. But this approach requires at least one true hypothesis at every time instant. This approach is not suitable for model interpolation as it requires the truth be presented in the model set.

In Chapter 3, we propose an algorithm which does not suffer with all these issues addressed above. The proposed algorithm uses Dirichlet distribution with Quasi-Bayes (QB) procedure for mixture estimation to estimate the model weights. This approach is natural because, DD will make sure PM constraints (2.6) are included. Extra care not needed for PM constraints. The DD with QB approach will work together very well to estimate the posterior model weights. The new algorithm works with single KF on PM, which makes algorithm computationally very attractive.

2.2 Dirichlet Distribution (DD)

The multinomial distribution is a discrete distribution which gives the probability of choosing a given collection of n items from a set of r items with repetitions and the probabilities of each choice is given by $\mu_1, \mu_2, ..., \mu_r$. The Dirichlet distribution (DD) [16] is the conjugate prior of the parameters of the multinomial distribution. The conjugacy property ensure that if a Bayesian up-

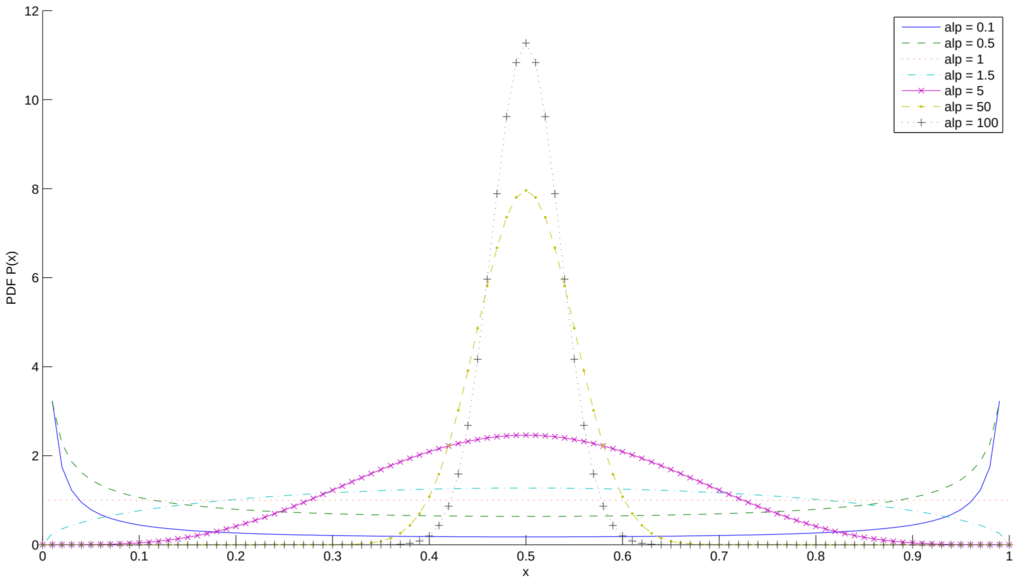

Figure 2.1: Two Dimensional Dirichlet PDF

date is performed with a Dirichlet prior, and a multinomially distributed sequence of observations, the updated distribution is still Dirichlet. Due to this nice property, the Dirichlet prior has been applied frequently in the Artificial Intelligence literature [26] [27] [28] [29].

The DD for $\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_r]'$ with parameters $\boldsymbol{\alpha} =[\alpha_1, \alpha_2, ..., \alpha_r]$, $(\alpha_i \geq 0)$, of order $r \geq 2$ is defined as

$$\mathcal{D}(\boldsymbol{\mu}; \alpha_1, ..., \alpha_r) = \frac{\Gamma(\alpha_1 + \alpha_2 + ... + \alpha_r)}{\Gamma(\alpha_1)\Gamma(\alpha_2)...\Gamma(\alpha_r)} \prod_{j=1}^{r} \mu_j^{\alpha_j - 1} \tag{2.9}$$

for

$$\Sigma_{j=1}^{r} \mu_j = 1,\ \mu_j \in [0, 1] \tag{2.10}$$

where Γ denotes the gamma function. DD can be viewed as multivariate Beta distribution [30], which is natural for $\boldsymbol{\mu}$ due to the unit simplex requirement. The parameters α_j can be interpreted as “prior observation counts” for events governed by μ_j.

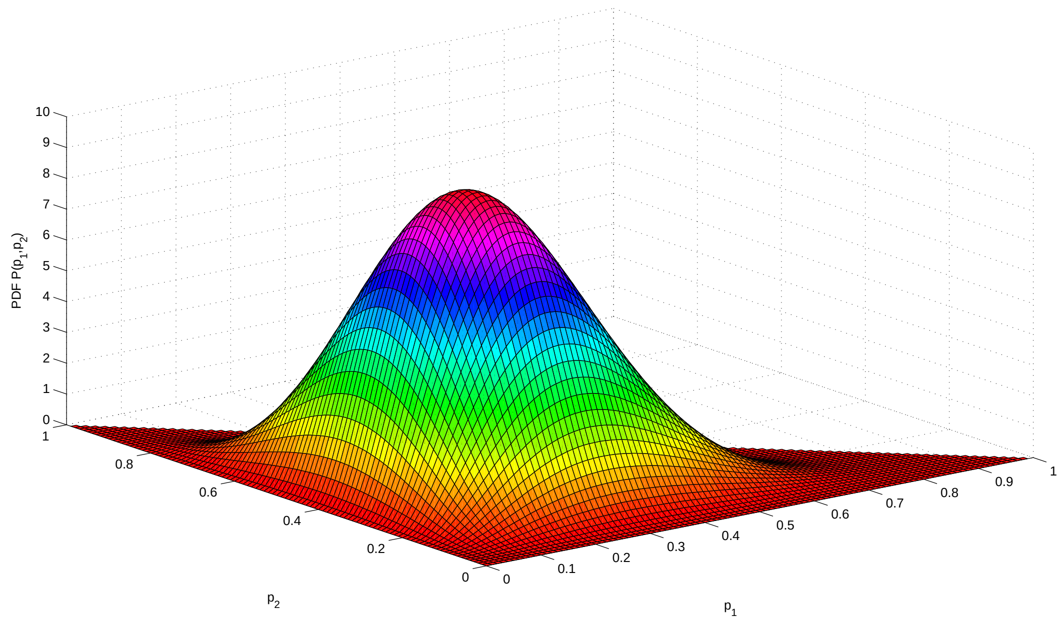

Figure 2.2: Three Dimensional Concave Dirichlet PDF

The mean and variance of the DD are

$$E[\mu_j] = \frac{\alpha_j}{\alpha_0}$$
$$Var[\mu_j] = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}$$

where $\alpha_0 = \sum_{i=1}^{r} \alpha_i$. The means of all μ_j stay the same if all α_j are scaled with the same multiplicative constant. The variances, however, get smaller as the parameters α grow.

DD offers a great variety of shapes and flexibility to model different situations through its parameters $[\alpha_1, \alpha_2, ..., \alpha_r]$. Figure 2.1 shows the two dimensional DD (beta distribution), when parameters α are assumed equal. It can be seen that as α is growing, distribution starts to change from concave to convex shape. Also it can be seen when parameters α are equal to unity, the distribution is a uniform distribution. Figures 2.2, 2.3 show three dimensional DD with α as parameter. Figure 2.2 shows DD when α_j are different. Note that the distribution is biased towards

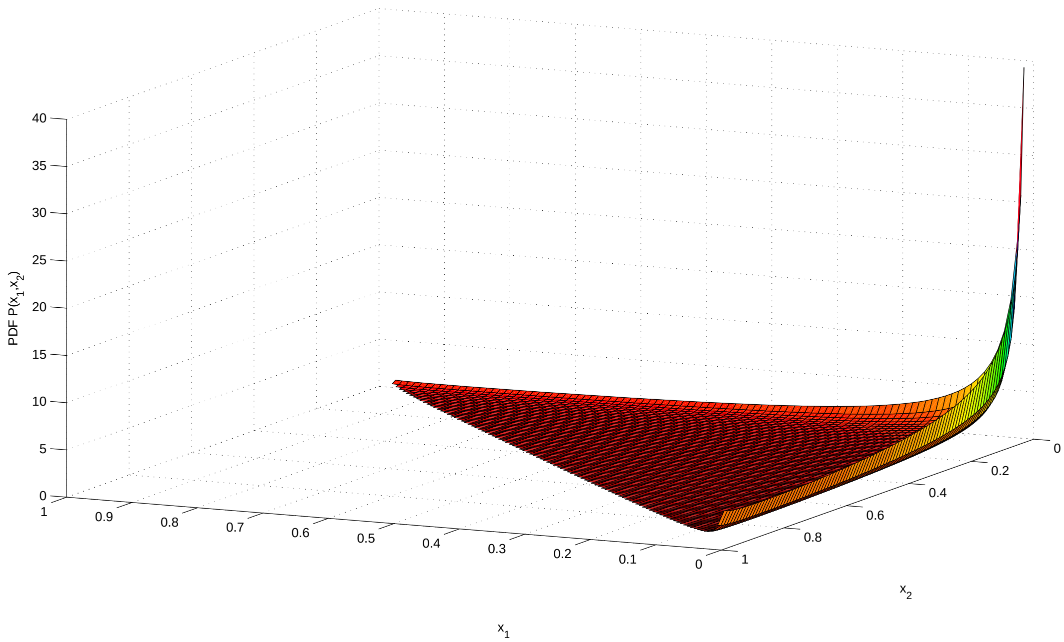

Figure 2.3: Three Dimensional Convex Dirichlet PDF

the larger value of α_j. In Figure 2.3, DD is plotted with different α_j but less than one. The figure illustrates that DD is convex in shape. So it can be concluded that when α_j are greater than one, DD is concave in shape otherwise convex in shape. The two figures illustrate that DD is biased with the α-vector.

DD Random Numbers Generation. If $Y_i, i = 1, 2, ..., r$ are i.i.d. random variables, following a Gamma distribution

$$Y_i \sim \Gamma(shape = \alpha_i, scale = \theta)$$

then

$$V = \sum_{i=1}^{r} Y_i \sim \Gamma(shape = \sum_{i=1}^{r} \alpha_i, scale = \theta)$$

and

$$X = [X_1, \ldots, X_r] = [X_1/V, \ldots, X_r/V] \sim \mathcal{D}(\alpha_1, \ldots, \alpha_r)$$

The above DD random number generation method is used later (Chapter 4) for sampling within the PM marginalized particle filter.

2.3 Quasi-Bayes (QB) Procedure for Mixture Probability Estimation

This section presents an approach to limit the computation/memory increase of the Bayesian mixture probability estimator with the help of Dirichlet prior. It is based on the so called Quasi-Bayesian (QB) approximation for estimation of finite mixtures [17]. The QB procedure is a nice choice for prior probability estimation (PPE) of finite mixtures (See [31], [32], [33] for other approaches) since it provides a quasi-posterior mean of the prior probabilities. It will be used later (see Chapter 3) to derive a QB filter for Polytopic models. The QB learning approach was also utilized for other applications, including transition probability estimation in hybrid system models [18], and adaptation of hidden Markov models (HMM) within an EM estimation framework for speech recognition [33] [34] [35] [36].

Consider probability density functions $f_1, f_2, ..., f_r$, conditional on $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_r)$. Assume random variables $X_n,\ n = 1, 2, \ldots$ are independent, with probability densities

$$p(x_n|\boldsymbol{\mu}) = \mu_1 f_1(x_n) + \mu_2 f_2(x_n) + ... + \mu_r f_r(x_n)$$

where all μ_i are non-negative and sum up to unity. The density f_i specifies the probability distribution of the observation, given that it belongs to population H_i, and μ_i denotes the probability of this event. Suppose $p(\boldsymbol{\mu})$ denotes a prior density for $\boldsymbol{\mu}$, $p(\boldsymbol{\mu}|x^n) = p(\boldsymbol{\mu}|x_1, x_2, ..., x_n)$ denotes the resulting posterior density for $\boldsymbol{\mu}$ given x^n. Also $p_i(\boldsymbol{\mu}|x^n)$ denotes the posterior density for $\boldsymbol{\mu}$ if, in addition to x^n, it were also known that the n^{th} observation came from H_i.

By Bayes theorem, for $n \geq 1$,

$$p(\boldsymbol{\mu}|x^n) \propto p(x^n|\boldsymbol{\mu})p(\boldsymbol{\mu}|x^{n-1})$$

Now defining RV's $Y_1, Y_2, ..., Y_n$, such that $Y_n = i$ if and only if X_n belongs to $H_i, i = 1, 2, ..., r$, then from above

$$p(\boldsymbol{\mu}|x^n) = \sum_{i=1}^{r} p(Y_n = i|x^n)p_i(\boldsymbol{\mu}|x^n) \tag{2.11}$$

The QB approximation relies on the assumption of DD [16] for the prior probabilities of the mixture (2.11).

After observing a measurement, say, x_1 from the mixture (2.11), the Dirichlet prior $p(\boldsymbol{\mu})$ yields a posterior $p(\boldsymbol{\mu}|x_1)$, which is a weighed sum of r DDs. Such a splitting leads in time to a weighted sum of an ever-increasing number of DD components. By assuming $p(\boldsymbol{\mu}) = \mathcal{D}(\boldsymbol{\mu};\alpha_1^{(0)}, \alpha_2^{(0)}, ..., \alpha_r^{(0)})$, after observing x_1,

$$p(\boldsymbol{\mu}|x_1) = \sum_{i=1}^{r} p(Y_1 = i|x_1)\mathcal{D}(\boldsymbol{\mu}; \alpha_1^{(0)} + \delta_{i1}, \alpha_2^{(0)} + \delta_{i2}, ..., \alpha_r^{(0)} + \delta_{ir})$$

where $p(Y_1 = i|x_1) \propto f_i(x_1)\alpha_i^{(0)}$ and $\delta_{ij} = 0$ if $i \neq j, \delta_{ij} = 1$ if $i = j$. The above posterior density is a linear combination of k DD's, could itself be approximated by a single DD. Then

$$p(\mu|x_1) = \mathcal{D}(\boldsymbol{\mu}; \alpha_1^{(0)} + \Delta_{11}, \alpha_2^{(0)} + \Delta_{12}, ..., \alpha_r^{(0)} + \Delta_{1r})$$

where $\Delta_{1i} = 1$ if x_1 belongs to $H_i, \Delta_{1i} = 0$ otherwise.

If the true population is not informed, then we have $\Delta_{11} = p(Y_i = i|x_1)$. Subsequent updating takes place entirely within the Dirichlet distribution of families; $p(\boldsymbol{\mu}|x_n)$ is DD with parameters

$$\alpha_i^{(n)}= \alpha_i^{(n-1)}+p(Y_n= i|x^n)$$

Also it can be verified from standard properties of DD that the posterior mean for μ_i, after observing $x_1, x_2, ..., x_n$, is given by

$$\bar{\mu}_i^{(n)} = \frac{\alpha_i^{(n)}}{\alpha_0 + n} \tag{2.12}$$

where $\alpha_0 = \alpha_1^{(0)} + \alpha_2^{(0)} + ... + \alpha_r^{(0)}$.

This idea of the above QB procedure is similar to the idea of the first-order generalized pseudo-Bayesian (GPB1) approach for MM state estimation [25], where at each time-step, the branching weighted sum of Gaussian's is approximated by a single Gaussian. .

Chapter 3

Proposed Methods for PM Estimation

3.1 QB Adaptive Kalman Filter

This part of thesis presents an algorithm for model weights estimation for PM (2.7)-(2.8) by using QB procedure explained in Chapter 2. The new algorithm, Quasi-Bayesian adaptive Kalman filter (QBAKF), estimates the model weights accurately by assuming weights are random and follow DD. The assumption of DD for model weights ensures that convex constraints for PM (2.6) are included naturally because DD implicitly accounts for the unit sum constraints, and positivity constraints. Further QB procedure for mixture probability estimation together with DD is perfect choice for model weights because of the conjugate property of DD. As explained in Chapter 2, with the QB procedure, the subsequent posterior update takes place within the family of DD's, when prior is assumed as DD. This property ensures straightforward calculation of quasi-posterior means $\bar{\mu}_i(k)$, $i = 1, 2, ..., r$.

The quasi-posterior mean update (2.12) is done directly through updating parameters α_i. The α_i update is done depending on how likely that model is going to happen. So calculating likelihood for each parameter will be challenging and varies depending on the problem.

The likelihood for QBAKF is calculated depending on the basis of bias at each time instant.

The bias is measured between the mixture calculated with previous step estimated model weights and individual components of the mixture similar to [8]. The bias is directly proportional to the mean of mixture components, so it leads to higher weights in the mixture. This larger model weight implies that truth is more inclined towards the corresponding model.

Upon receiving each measurement, single Kalman Filter (KF) is implemented on the model mixture with the old weights. Then the likelihood of each model is calculated by approximating the innovation parameter being normally distributed with mean as bias and covariance as innovation covariance from KF. Further model weights are updated through likelihood and used for next time cycle. The appealing thing about this approach is, it uses only one KF on the mixture at every time instant. The QBAKF is computationally efficient compared to other approaches because of single KF. It interpolates well between the local models in the model set. It does not require that the truth is one of the local models in the multiple model set. The QBAKF can be extended to maneuvering problems, but it is slow to reach the truth. It can be improved by proper modeling of the weights for maneuvers. In the next section, QBAKF development and description are provided.

3.1.1 Derivation

This part explains in detail, how QBAKF algorithm is developed. The QBAKF utilizes QB procedure and bias to estimate the mixture weights and then state vector. In the following QBAKF is derived for the problem formulated in (2.7)-(2.8).

As explained in the previous section, estimating weights are done through likelihood of each parameter. This section will explain in detail, how likelihood is calculated and then weights are estimated. The weight for each model is calculated by using QB procedure. The main goal here is to find out the bias of each model and then by calculating likelihood of each model $g_i(k)$ with the bias. From the Section 2.3, the quasi-posterior mean of the weight for i^{th} model is given by the following algorithm,

$$\alpha_i(k) = \alpha_i(k-1) + \frac{\alpha_i(k-1)g_i(k)}{\sum_{l=1}^{r}\alpha_l(k-1)g_l(k)} \tag{3.1}$$

$$\bar{\mu}_i(k) = \frac{\alpha_i(k)}{k+\alpha(0)} \tag{3.2}$$

where k is the time sample, $\alpha_i(0) > 0$, and $\alpha(0) = \sum_{i=1}^{r}\alpha_i(0)$.

Consider system equations are for i^{th} model from (2.4)–(2.5),

$$\begin{aligned} x_k &= F_k^{(i)}x_{k-1} + G_k^{(i)}u_{k-1} + w_k \\ z_k &= H_k^{(i)}x_k + v_k \end{aligned}$$

where $(F_k^{(i)}, G_k^{(i)}, H_k^{(i)})$ corresponds to i^{th} model.

Let $F_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{F}_k$, $G_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{G}_k$, $H_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{H}_k$ and $\hat{x}_k$, P_k denotes the estimate of the KF with F_k, G_k, H_k.

Consider also a hypothetical one step KF prediction part for i^{th} model. So the prediction equations for model i are

$$\begin{aligned} \bar{x}_k^i &= F_k^{(i)}\hat{x}_{k-1} + G_k^{(i)}u_{k-1} \\ \bar{z}_k^{(i)} &= H_k^{(i)}\bar{x}_k^i \\ &= H_k^{(i)}(F_k^{(i)}\hat{x}_{k-1} + G_k^{(i)}u_{k-1}) \\ &= H_k^{(i)}(\Delta F_k^{(i)}\hat{x}_{k-1} + \Delta G_k^{(i)}u_{k-1}) + H_k^{(i)}(F_k\hat{x}_{k-1} + G_k u_{k-1}) \\ &= H_k^{(i)}(\Delta F_k^{(i)}\hat{x}_{k-1} + \Delta G_k^{(i)}u_{k-1}) + (\Delta H_k^{(i)} + H_k)(F_k\hat{x}_{k-1} + G_k u_{k-1}) \\ &= H_k^{(i)}(\Delta F_k^{(i)}\hat{x}_{k-1} + \Delta G_k^{(i)}u_{k-1}) \\ &\quad + \Delta H_k^{(i)}(F_k\hat{x}_{k-1} + G_k u_{k-1}) + H_k(F_k\hat{x}_{k-1} + G_k u_{k-1}) \end{aligned}$$

where $\bar{x}_k^i$ is the predicted state, $\bar{z}_k^{(i)}$ is the predicted measurement, and

$$\Delta F_k^{(i)} = F_k^{(i)} - F_k,\ \Delta G_k^{(i)} = G_k^{(i)} - G_k,\ \Delta H_k^{(i)} = H_k^{(i)} - H_k$$

Next, by manipulating above predicted measurement equations the following relations are obtained.

$$\begin{aligned}
\bar{z}_k^{(i)} &= \bar{z}_k + (H_k^{(i)}\Delta F_k^{(i)} + \Delta H_k^{(i)} F_k)\hat{x}_{k-1} + (H_k^{(i)}\Delta G_k^{(i)} + \Delta H_k^{(i)} G_k)u_{k-1} \\
\tilde{z}_k &= \tilde{z}_k^{(i)} + (H_k^{(i)}\Delta F_k^{(i)} + \Delta H_k^{(i)} F_k)\hat{x}_{k-1} + (H_k^{(i)}\Delta G_k^{(i)} + \Delta H_k^{(i)} G_k)u_{k-1}
\end{aligned}$$

where $\tilde{z}_k = z_k - \bar{z}_k$, $\tilde{z}_k^{(i)} = z_k - \bar{z}_k^{(i)}$, z_k is measurement, and $\bar{z}_k$ is predicted measurement for the mixture model.

Now likelihood for i^{th} model can be approximated depending on the bias, similarly to [8], by

$$g_i(k) = \mathcal{N}(z_k - \hat{z}_{k|k-1}; b_k^{(i)}, S_k)$$

where $b_k^{(i)} = (H_k^{(i)}\Delta F_k^{(i)} + \Delta H_k^{(i)} F_k)\hat{x}_{k-1} + (H_k^{(i)}\Delta G_k^{(i)} + \Delta H_k^{(i)} G_k)u_{k-1}$, z_k is measurement, $\hat{z}_{k|k-1}$ is the predicted value of the measurement, S_k is innovation covariance. Here likelihood is assumed to follow Gaussian distribution with parameter as innovation at each time step with mean $b_k^{(i)}$ and covariance as innovation covariance from the KF. The approximation is for one time step of KF, however, it can be extended to multiple steps.

Upon calculating the likelihoods of each model, model weights are calculated using QB procedure mentioned above. These weights are used for next time cycle. For the next time cycle again single KF is implemented on the mixture of models with these weights. This cycle will repeat. Because of natural convex property of DD, weights will follow summing to one and positivity constraints. The next section outlines the QBAKF algorithm.

3.1.2 QBAKF Algorithm

- A priori parameters:

  $\boldsymbol{\alpha}(0) = [\alpha_1(0), \alpha_2(0), ..., \alpha_r(0)]'$

  $\alpha(0) = \sum_{i=1}^{r} \alpha_i(0), \qquad \alpha_i(0) > 0, \qquad i = 1, 2, ..., r;$

- Initialization:

  $\bar{\boldsymbol{\mu}}(0) = \frac{1}{\alpha(0)}\boldsymbol{\alpha}(0)$, $\hat{x}_{0|0}$, $P_{0|0}$

- For $k = 1, 2, ...$

  –State Estimation:

  $F_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{F}_k$, $G_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{G}_k$, $H_k = \bar{\boldsymbol{\mu}}(k-1)\mathbf{H}_k$

  $[\hat{x}_{k|k}, P_{k|k}, z_{k|k-1}, S_k] = \mathbf{KF}\ [\hat{x}_{k-1|k-1}, P_{k-1|k-1}, z_k, F_k, G_k, H_k]$

  –Weights Estimation:

  For $i = 1, 2, ..., r;$

  $\Delta F_k^{(i)} = F_k^{(i)} - F_k$, $\Delta G_k^{(i)} = G_k^{(i)} - G_k$, $\Delta H_k^{(i)} = H_k^{(i)} - H_k$

  $b_k^{(i)} = (H_k^{(i)}\Delta F_k^{(i)} + \Delta H_k^{(i)} F_k)\hat{x}_{k-1} + (H_k^{(i)}\Delta G_k^{(i)} + \Delta H_k^{(i)} G_k)u_{k-1}$

  $g_i(k) = \mathcal{N}(z_k - z_{k|k-1}; b_k^{(i)}, S_k)$

  $\alpha_i(k) = \alpha_i(k-1) + \frac{\alpha_i(k-1)g_i(k)}{\sum_{l=1}^{r}\alpha_l(k-1)g_l(k)}$

  $\bar{\mu}_i(k) = \frac{1}{k+\alpha(0)}\alpha_i(k)$

  $\bar{\boldsymbol{\mu}}(k) = [\bar{\mu}_1(k), \bar{\mu}_2(k), ..., \bar{\mu}_r(k)]$

The necessary parameter vector $\boldsymbol{\alpha}(0)$ represent the unnormalized a priori model weight vectors $\bar{\boldsymbol{\mu}}(0)$ and are normalized, so that the initial model weight estimate belongs to the unit simplex of valid stochastic matrices.

Regarding the initialization of the algorithm, the following property of the DD is very useful in practical application: If its parameters are chosen as α_1 = α_2 = ...= α_r = 1,, it coincides with the uniform distribution over the unit $(r-1)$- dimensional simplex. When no a priori information about the model weights is unavailable, the QBAKF estimator is initialized with the noninformative (uniform) prior, $\bar{\mu}_i(0) = 1/r, i = 1, 2, ..., r$. The QBAKF algorithm is extremely simple for implementation and requires very less computation. The convergence properties of the QB approach are discussed in [17] and the relevant references therein.

3.2 DD Marginalized Particle Filter

3.2.1 Motivation

The estimation problem (2.7)-(2.8) is nonlinear and non-Gaussian, and particle filter (PF) is a viable option to apply. However generic PF is not a good choice, because for large number of models particle state dimension is $n_x + r$ is large and many particles will be needed. Asymptotically as the number of particles tends to infinity we know that we get the optimal filter. An inherent problem with the PF is its high computational cost. In practice there is a tradeoff between accuracy and computational complexity. The state dimension is biggest limiting factor for PF approximation, when it is large.

Given model weight vector $\boldsymbol{\mu}$, the model (2.7)-(2.8) is linear, and Rao-Blackwelization technique [19] can be applied. That is, perform linear KF for the linear part given $\boldsymbol{\mu}$, and perform PF for the nonlinear part with respect to $\boldsymbol{\mu}$ only. This could save a tremendous amount of computation and make the filter feasible. This method is like marginalizing the linear state variables and estimating using KF similar to marginalized particle filter (MPF) [20].

3.2.2 DDMPF Algorithm

This section presents a new algorithm for PM estimation, referred to as a DD Marginalized PF (DDMPF). The main assumption is that, the weight vectors in the problem (2.7)-(2.8) are random and obeys DD [17]. The DD makes weights naturally to follow convex property. The importance sampling step for model weights is according to

$$\boldsymbol{\mu}^{(i)}(k) \sim p(\boldsymbol{\mu}(k)|\boldsymbol{\mu}^{(i)}(k-1)) = \mathcal{D}(\boldsymbol{\mu}(k); \boldsymbol{\mu}^{(i)}(k-1) + w_{\boldsymbol{\mu}}^{(i)}(k))$$

where $w_{\boldsymbol{\mu}(k)}$ is zero mean white Gaussian noise with small covariance. There is no external adjustment needed for the weights to follow convex property.

In the standard particle filters, obtained variance of the estimates can be decreased by exploiting linear substructures in the model. The main idea behind MPF is to marginalize corresponding variables and estimate using an optimal linear filter.

The DDMPF algorithm has two steps, sampling and resampling. In the sampling step new weights $\boldsymbol{\mu}(k)$ are estimated by using DD with the parameter as old weights $\boldsymbol{\mu}(k-1)$ with zero mean white Gaussian noise is added. The problem becomes linear estimation problem because weights of the models are known. After by using M linear KFs, correspondingly M state vectors are estimated. First step of the algorithm is concluded by calculating probability masses of the each filter. In the second step, regular resampling is done.

The DDMPF and standard particle filter are closely related. The DDMPF is given below for one time cycle.

DDMPF Algorithm

- Initialization:

  For $i = 1, ..., M$

  $$\{\hat{x}_{0|0}^{(i)}, P_{0|0}^{(i)}, \boldsymbol{\mu}^{(i)}(0)\} = \{\bar{x}_0, P_0, \bar{\boldsymbol{\mu}}(0)\}$$

- Sampling:

  For $i = 1, ..., M$

  $$\boldsymbol{\mu}^{(i)}(k) \sim \mathcal{D}(\boldsymbol{\mu}^{(i)}(k); \boldsymbol{\mu}^{(i)}(k-1) + w_{\boldsymbol{\mu}(k)}^{(i)})$$

  $$\mathcal{F}_k^{(i)} = \boldsymbol{\mu}^{(i)}(k)\mathbf{F}_k, \mathcal{G}_k^{(i)} = \boldsymbol{\mu}^{(i)}(k)\mathbf{G}_k, \mathcal{H}_k^{(i)} = \boldsymbol{\mu}^{(i)}(k)\mathbf{H}_k$$

  $$[\hat{x}_k^{(i)}, P_k^{(i)}, \hat{z}_{k|k-1}^{(i)}, S_k^{(i)}] = \mathbf{KF}[\hat{x}_{k-1}^{(i)}, P_{k-1}^{(i)}, z_k, \mathcal{F}_k^{(i)}, \mathcal{G}_k^{(i)}, \mathcal{H}_k^{(i)}]$$

  Importance weights:

  $$\tilde{q}_k^{(i)} = \mathcal{N}(z_k; \hat{z}_{k|k-1}^{(i)}, S_k^{(i)})$$

- Normalization:

  For $i = 1, ..., M$

  $$q_k^{(i)} = \frac{\tilde{q}_k^{(i)}}{\sum_{i=1}^M \tilde{q}_k^{(i)}}$$

- Resampling:

  For $i = 1, ..., M$

  $$P\{\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(j)}\} = q_k^{(j)}$$

- Output:

  $$\hat{x}_{k|k} = \sum_{i=1}^M q_k^{(i)} \hat{x}_k^{(i)} \qquad P_{k|k} = \sum_{i=1}^M q_k^{(i)} (P_k^{(i)} + (\hat{x}_{k|k} - \hat{x}_k^{(i)})(\hat{x}_{k|k} - \hat{x}_k^{(i)})')$$

The DDMPF algorithm is quite general and further improvements can be made. Initialization of model weights can be improved and the evolution of weights also can be improved depending on the nonlinear problem. The number of KFs (M) can be chosen depending on the nonlinearity of the problem.

Chapter 4

Simulation Study

In this chapter, two target tracking examples are considered. The unknown parameters are the sampling period and the control input matrix, respectively. For the two examples, the tracking problem of a non-maneuvering target in the presence of noisy measurements is considered. The goal of tracking target is to obtain a consistent estimate of the state even in case of noisy measurements. The target tracking problem considered in this section is based on the tracking problem considered in [24] [19] [38]. The state of target is given by $x = (\mathsf{x}, \dot{\mathsf{x}}, \mathsf{y}, \dot{\mathsf{y}})'$, where x and y represent the positions in the x and y directions of the target and ẋ and ẏ represent the velocities in the x and y directions, respectively. The non-maneuvering target evolves according to a JMLS [39].

$$x_{k+1} = Fx_k + Gu_k + w_k \tag{4.1}$$

$$z_k = Hx_k + v_k \tag{4.2}$$

where $k = 0, 1, \ldots$ is the time index.

The initial state is random and normally distributed, i.e., $x_0 \sim \mathcal{N}(\bar{x}_0, P_0)$. The Gaussian process and measurement noises, $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, R)$, respectively, are white and mutually uncorrelated.

The matrices, state-transition (F), input-control (G) and measurement (H) are defined below to model different modes of motion. For the two examples, target motion (4.1) is assumed to follow nearly Constant-Velocity (CV) motion. The CV model is intended to describe the non-maneuvering mode of motion.

For each tracking filter the following performance measures are computed by Monte Carlo simulation with M realizations (runs). For more details regarding evaluating the performance of estimators the reader is referred to [40].

The accuracy of the algorithms is measured in terms of root-mean-square errors (RMSE):

Position RMSE:

$$PRMSE_k = \left(\frac{1}{M} \sum_{i=1}^{M} [(\mathsf{x}_k^{(i)} - \hat{\mathsf{x}}_k^{(i)})^2 + (\mathsf{y}_k^{(i)} - \hat{\mathsf{y}}_k^{(i)})^2] \right)^{1/2}$$

Velocity RMSE:

$$VRMSE_k = \left(\frac{1}{M} \sum_{i=1}^{M} [(\dot{\mathsf{x}}_k^{(i)} - \hat{\dot{\mathsf{x}}}_k^{(i)})^2 + (\dot{\mathsf{y}}_k^{(i)} - \hat{\dot{\mathsf{y}}}_k^{(i)})^2] \right)^{1/2}$$

where $x_k = (\mathsf{x}_k^{(i)}, \mathsf{y}_k^{(i)}, \dot{\mathsf{x}}_k^{(i)}, \dot{\mathsf{y}}_k^{(i)})'$, $\hat{x}_k = (\hat{\mathsf{x}}_k^{(i)}, \hat{\mathsf{y}}_k^{(i)}, \hat{\dot{\mathsf{x}}}_k^{(i)}, \hat{\dot{\mathsf{y}}}_k^{(i)})'$ denote the true and estimate states, respectively, in run i at time k.

For the simulations, process and measurement noise covariances are,

$$Q = 10^{-3} I_4, \; R = \begin{bmatrix} 2500 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2500 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The sensor is located at $(0, 0)$. The noise-free target initial location is at $(-500m, -500m)$ and the noise-free initial speed is $5m/s$. Both the state and measurement vectors are in Cartesian coordinates. The system is simulated for 250 samples. In this simulation non-maneuvering target is considered over sampling interval [1, 250]. The state estimation results are obtained with 100 Monte-Carlo (MC) simulations for two examples.

4.1 Example 1 (Unknown Sampling Interval)

In this example, true sampling interval is assumed to be unknown and assumed to belong to a known interval. The boundaries of the interval are considered as models with certain initial model weights. Now the problem can be converted into PM. The PM with the different filters is used to estimate the weights of those models, so that true sampling interval can be estimated.

The model is given by (4.1) with $u_k = 0$ and system matrices

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; G = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}, \; H = I_4$$

are considered. Where T is the sampling time interval. So the switching term in the model (4.1) is F matrix. For the simulation purpose T is assumed to interpolate between two values that correspond to two different sampling periods. These two matches are given by,

$$T_1 = 1s, \; T_2 = 10s$$

For generating the truth, sampling interval $T = 7s$ is simulated over the time interval [1, 250]. As explained in Chapter 2, one of the benefits with the PM structure is interpolation property. To illustrate this property, QBAKF and DDMPF are implemented with two models $T_1 = 1s$ and $T_2 = 10s$. The DDMPF is implemented with 100 particles.

The QBAKF and DDMPF are compared with the performances of EKF, AMM, and IMM. The EKF is implemented by augmented state consisting of the state x_k and the weight vector $\bar{\pi}(k)$.

Then the problem becomes nonlinear, so non-linear KF, i.e., EKF is implemented. The covariance matrices for model weight vector is considered as $Q_k^{\boldsymbol{\mu}} = 0.01 I_2$ for EKF, for all k.

The Autonomous multiple model (AMM) [21] with two models is implemented. The AMM algorithm runs a conditional Kalman filter (KF) for each model in the model set and evaluates the posterior probability of each model. The overall fused estimate is obtained as a sum of the conditioned estimates weighted by their corresponding model probabilities. The conditional filters operate independently, in an autonomous manner, no information exchanged among the filters.

The Interacting Multiple Model (IMM) [22] algorithm, runs several conditional filters. However, each filter is individually re-initialized with the estimate, refereed to as mixed estimate, conditioned on the model at the current time instant. After updating the filters IMM provides the overall output through estimate fusion, but maintains the conditional estimates for the next time step. The IMM in the present simulation uses two models. The transition probability matrix used for IMM is given by,

$$\Pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}$$

The initial model weights are assumed to be equal and follow convex property for all filters considered here for comparison. The covariance matrices are chosen same as Q and R for all filters, for all k.

Comparative time-plots of the position and velocity RMSEs are shown in Figures 4.1 and 4.2, respectively. It can be seen from the RMSE plots, QBAKF is outperforming all other filters by large margin. The DDMPF is also performing well but not as good as QBAKF. This is probably because of modeling weights and number of particles. The AMM is performing not really well, because with the AMM, probability of the model in the model set, with the smallest distance to the true model tends to unity almost surely as time increases. In this case, true mode is not presented in the model set. So AMM tends to provide the unity probability to the second model i.e T_2, because truth is closer to it. The AMM performance can be improved by increasing the number of valid

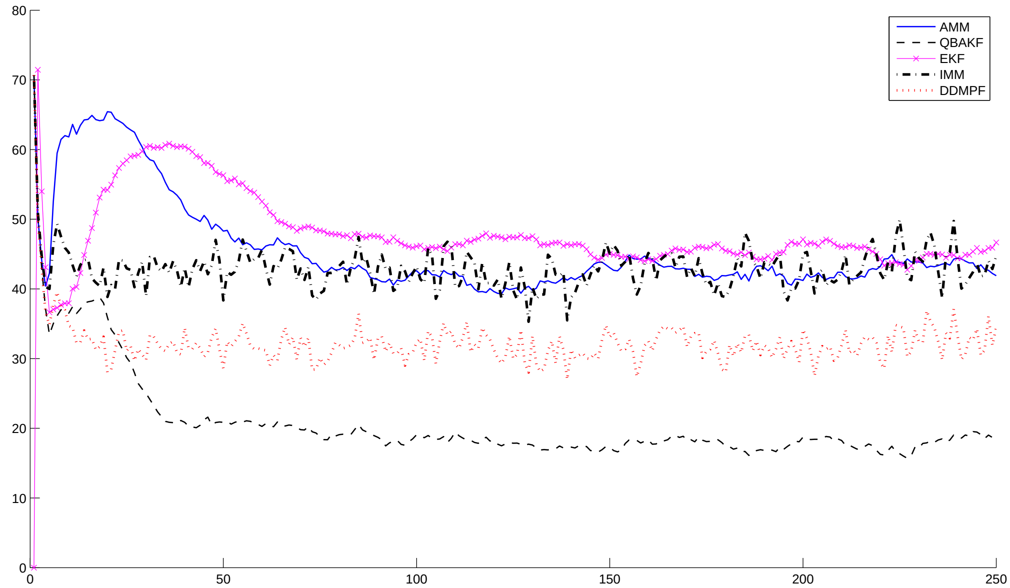

Figure 4.1: Position RMSE

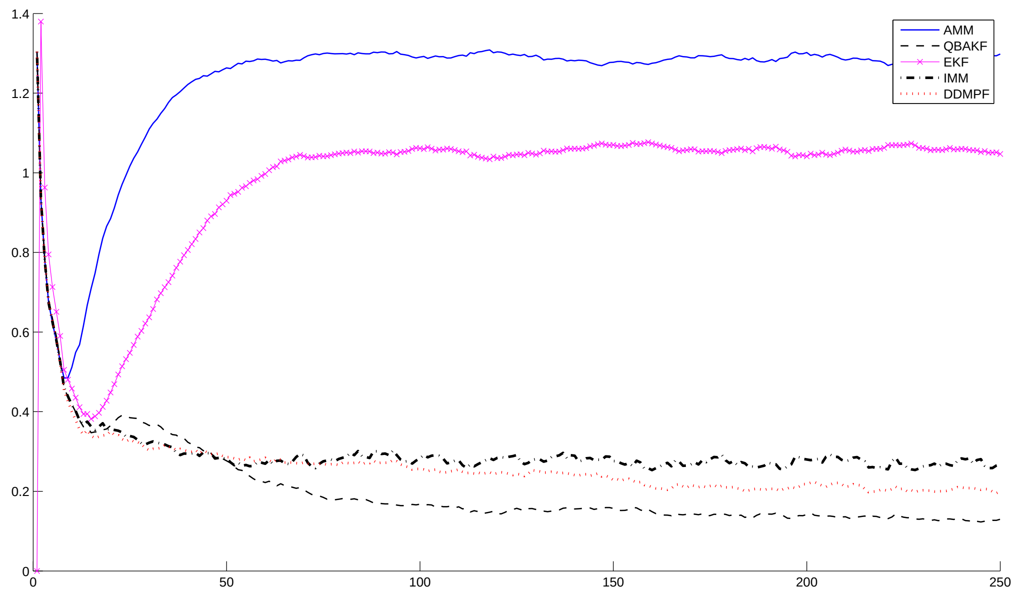

Figure 4.2: Velocity RMSE

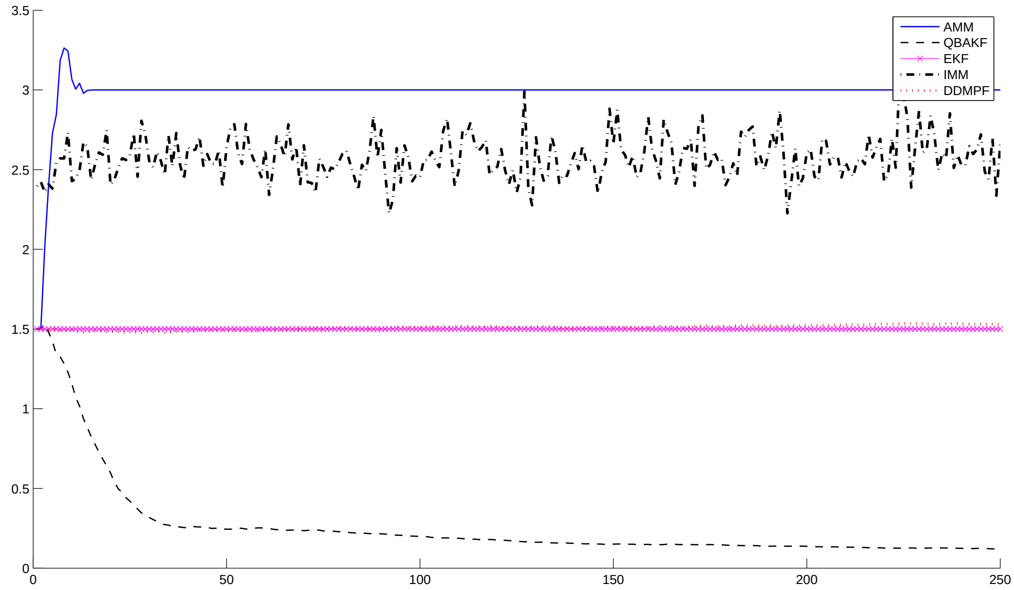

Figure 4.3: Sampling Time RMSE

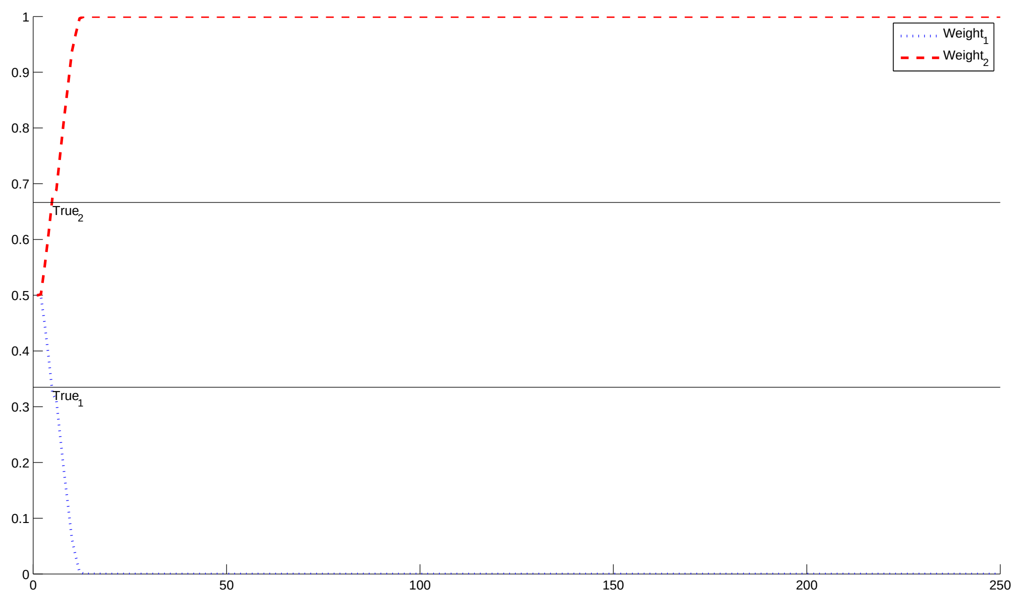

Figure 4.4: AMM Estimated Weights

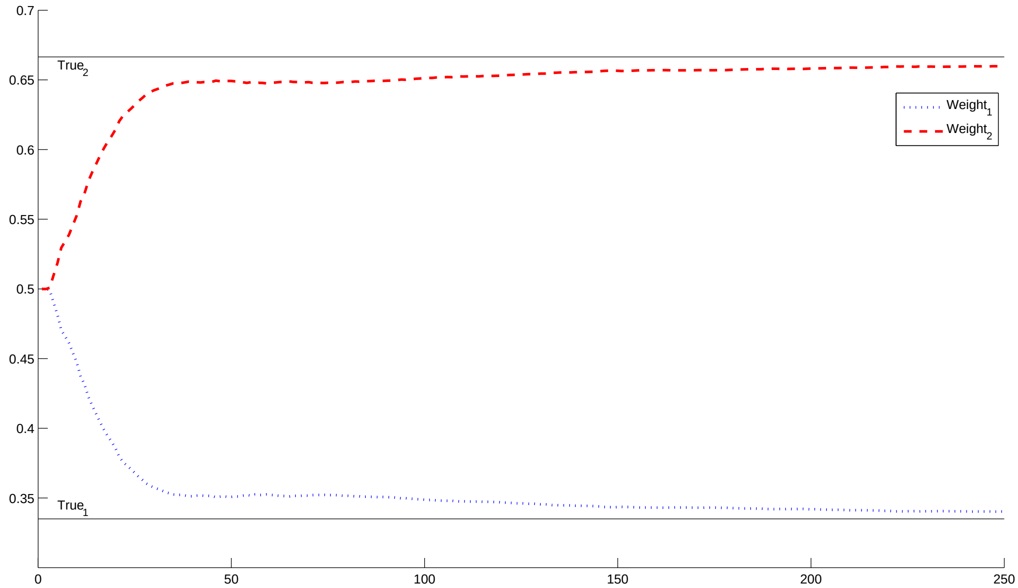

Figure 4.5: QBAKF Estimated Weights

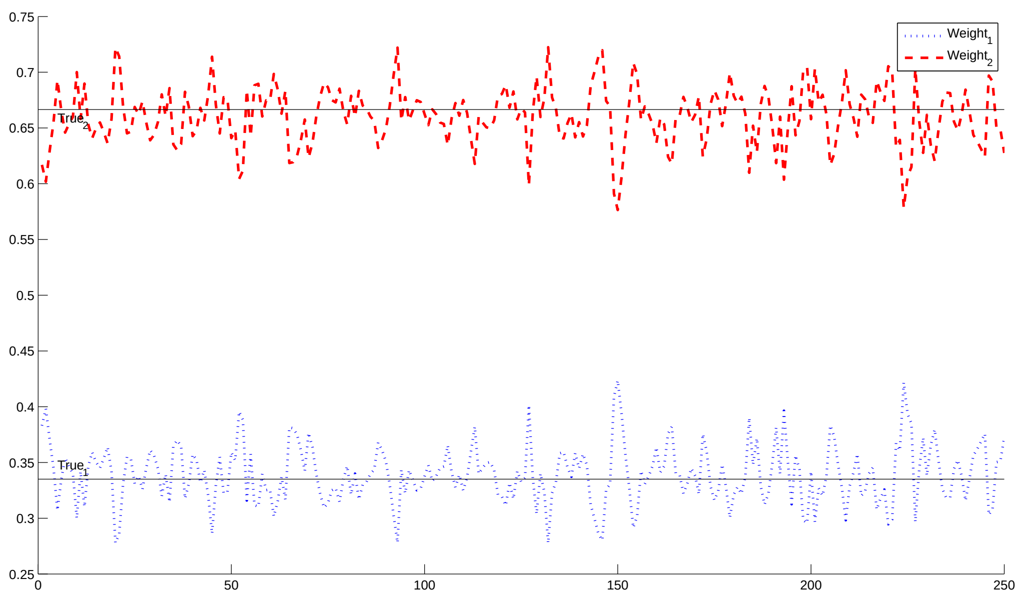

Figure 4.6: IMM Estimated Weights

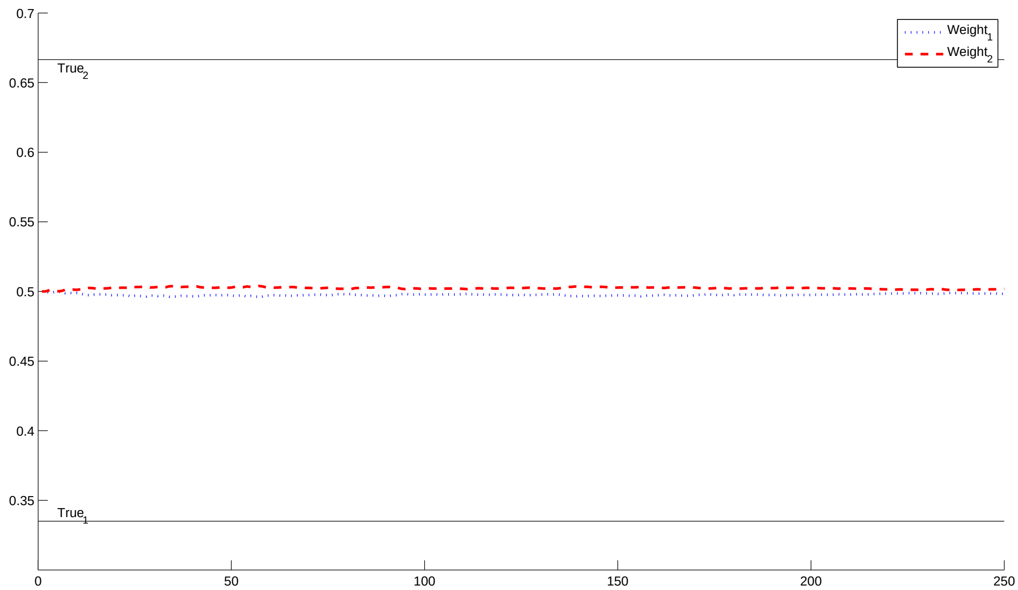

Figure 4.7: DDMPF Estimated Weights

models in model set. At the same time EKF is performing badly because model weights in EKF are assumed to follow random walk process. So most of the times model weights in EKF tries to be around initial weights as the time goes, which is not suitable for this kind of estimation. The IMM is also not very good in terms of accuracy of algorithms. It is because of the fundamental limitation of IMM with fixed structure i.e., IMM believes at any given time one of their filters is perfect and none of them may provide incorrect information. The filters in IMM trust themselves so much. In this case truth is not there in the IMM model set. Here IMM is exposed to the mode that is unknown.

Figure 4.3 illustrates accuracy of parameter estimation. Sampling interval RMSEs with respect to estimated T are compared. Again QBAKF is far better among all the algorithms because QBAKF is able to estimate weights very well. Other algorithms are not able to model weights properly because of their fundamental limitations for this kind of problem.

In Figures 4.4-4.7, estimated model weights of AMM, QBAKF, IMM and DDMPF are plotted respectively. The true weights are also presented. For the truth of $T = 7s$, model weight vector should be $(0.335, 0.6665)$. Again it can be seen, QBAKF is best estimating the weights towards the truth while others are not able to do.

4.2 Example 2 (Unknown Control Input matrix)

The second example, the switching term in the model (4.1) is G matrix. For this example, the model is given by (4.1) with $u_k = \sqrt{2}$ and sampling interval $T = 1s$. All remaining parameters are same as in the Example 1. To model G matrix, interpolation between three possible matrices is considered. Three possible values that correspond to three different maneuver commands. These three matrices are given by

$$\begin{aligned} G^{(1)} &= [-0.5, 0, 0, 1]' \\ G^{(2)} &= [-0.25, 0.5, 0.25, -0.5]' \\ G^{(3)} &= [0.25, -0.5, 0.25, -0.5]' \end{aligned}$$

The truth is simulated with $0.14G^{(1)} + 0.33G^{(2)} + 0.53G^{(3)}$ over the time interval [0, 250]. So the true model weight vector is $(0.14, 0.33, 0.53)$. The performances of EKF, AMM and IMM are compared with QBAKF. All the algorithms use three models with the equal initial weights. The transition probability matrix for IMM is considered as

$$\Pi = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}$$

In Figures 4.8 and 4.9 RMSEs of position and velocity are compared. The QBAKF is one of the algorithms with least RMSE. As time goes, RMSE of QBAKF is gradually decreasing

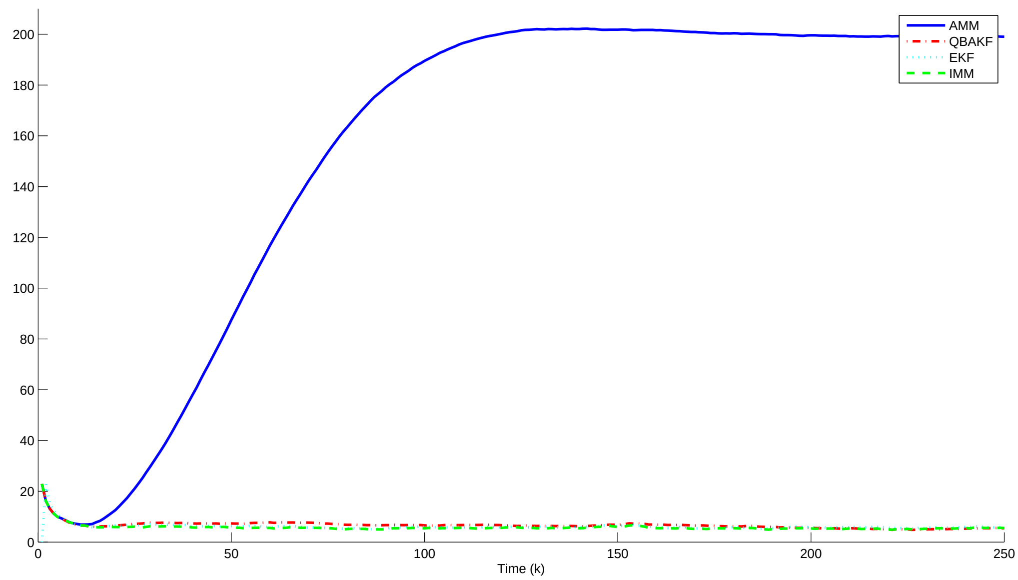

Figure 4.8: Position RMSE

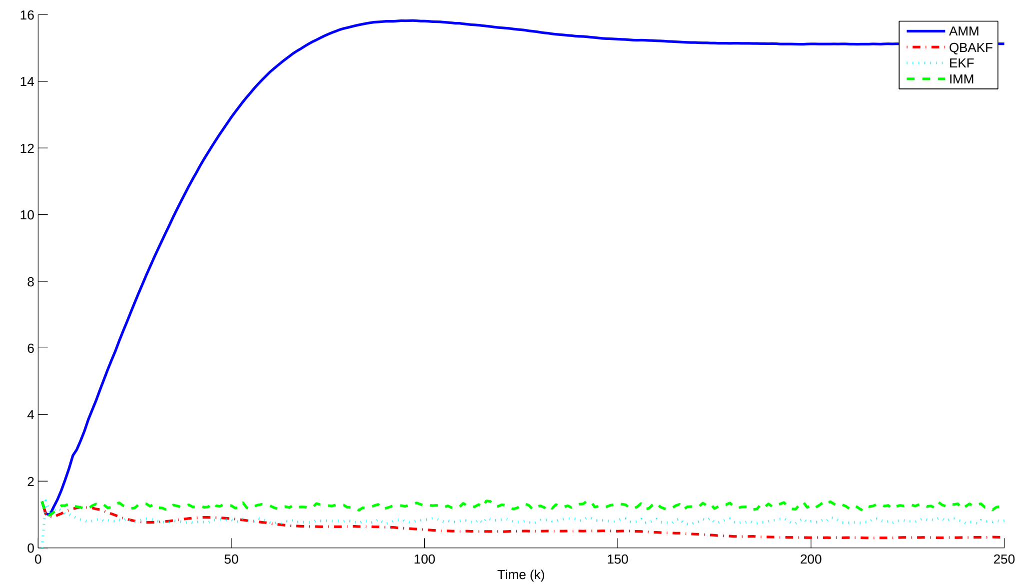

Figure 4.9: Velocity RMSE

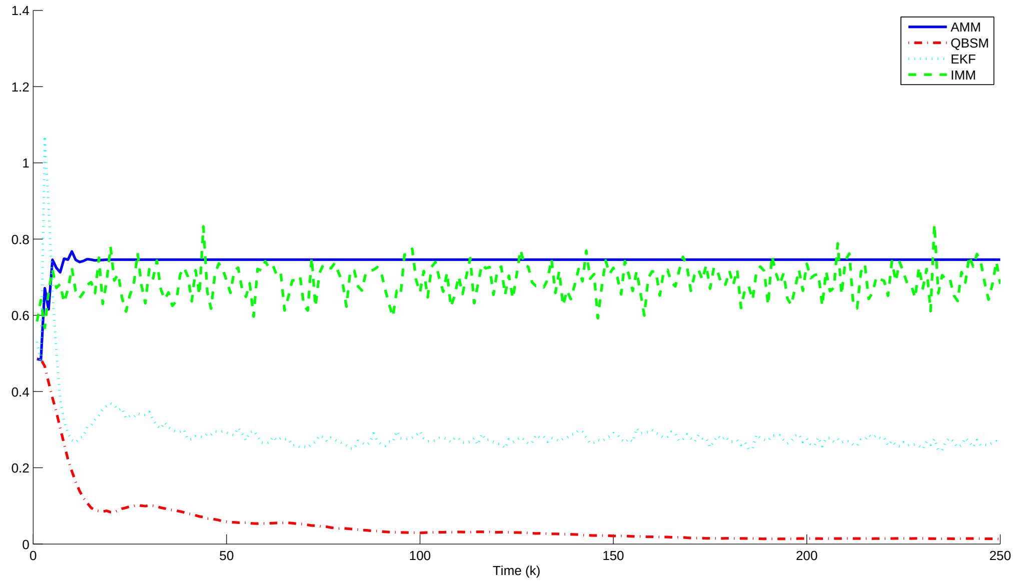

Figure 4.10: RMSE of Estimating Parameter G

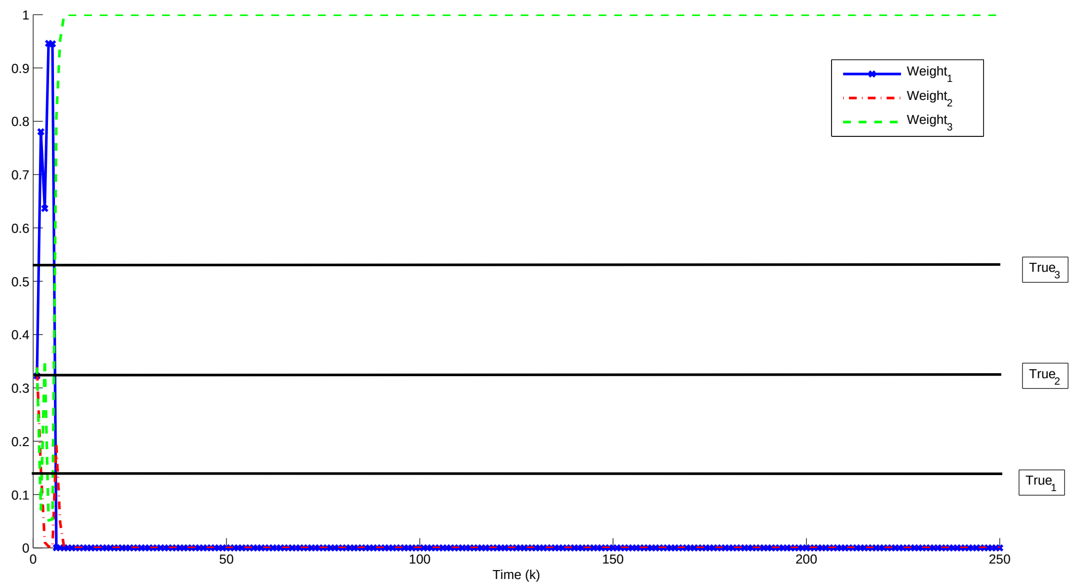

Figure 4.11: AMM Estimated Weights

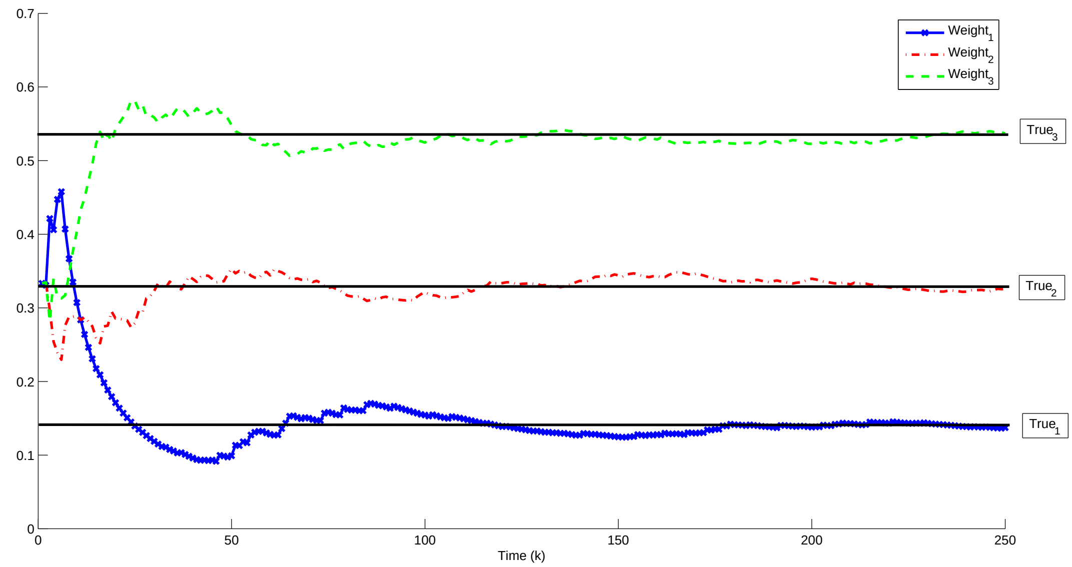

Figure 4.12: QBAKF Estimated Weights

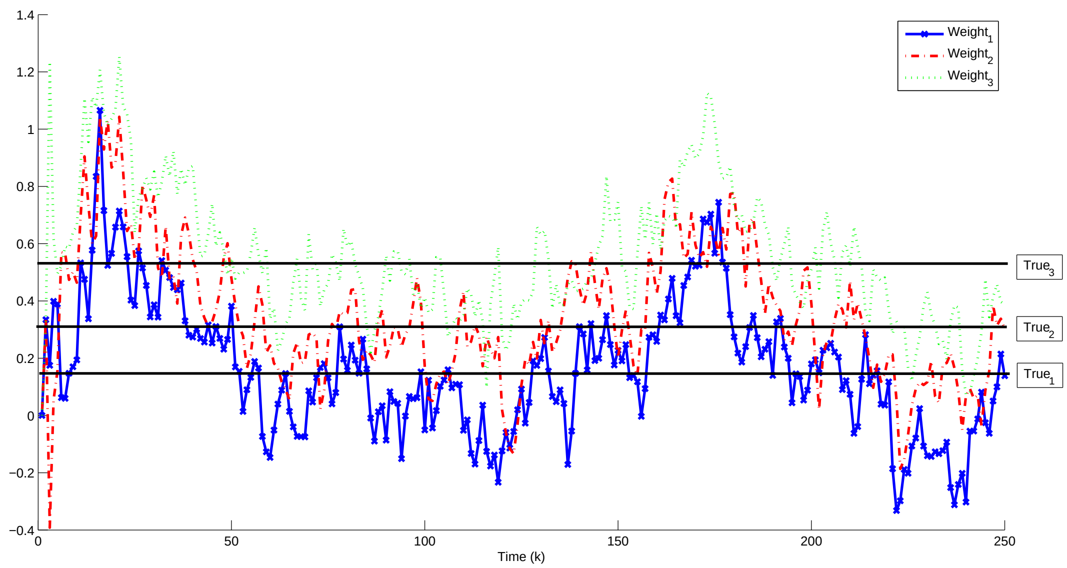

Figure 4.13: EKF Estimated Weights

when compared to all other algorithms. The Figure 4.10 is about RMSE of parameter G from the estimated model weights. Out of all algorithms, QBAKF is with almost perfect reconstructed G matrix. The Figures 4.11-4.13 show estimated model weights for AMM, QBAKF and EKF respectively. The true weights are also shown for comparison. The estimated EKF weights are not following convex property. This is one of the biggest disadvantage with augmented EKF to estimate model weights. Out of all those estimated model weights, QBAKF is with almost perfect estimated model weight vector.

An important conclusion that can be drawn from the above two examples is that the PM structure is preferred when interpolation between models from the model set is required. The regular algorithms EKF, AMM, and IMM will not work very well for the PM structures as because of their fundamental limitations/assumptions. The QBAKF performs far better for the interpolation of the models in the model set. The best thing about this algorithm is able to estimate the model weight vectors with minimum error. Also QBAKF runs only single KF for mixture and able to estimate the model weights properly. It is simple to implement when compared to other PM estimation algorithms. The only shortcoming for QBAKF is when the true mode is in the model set. Other algorithms like AMM will perform better. Apart from that, QBAKF can be used for the model sets with PM structure. The DDMPF algorithm simulated in the first example is also not doing bad, but it can be further improved by proper modeling and increasing number of particles. The main shortcoming is the high computational cost.

Chapter 5

Summary and Conclusions

In this thesis, an alternative model structure is adopted and utilized for the hybrid model structure of JMLS. This structure, named polytopic model (PM) structure, explicitly allows for interpolation between models. The estimation algorithms based on PM have better model interpolation properties. The PM structure is free of transition probabilities, which simplifies estimation process.

Two novel estimation algorithms, QBAKF and DDMPF, are proposed and implemented for PM estimation problems. In this thesis, model weights are assumed to be random and follow DD because of convex property of the model weights. The assumption of DD for model weights is quite natural, as model weights from the DD follow convex property automatically. The QBAKF algorithm uses Quasi-Bayesian approach for mixture probability estimation with the single KF for estimating model weights in the mixture. It is very simple to implement. The QBAKF is weakly dependent on the number of models in the model set, it runs only one KF on mixture.

The DDMPF algorithm is similar to MPF. The DDMPF estimates model weights by assuming, weight vectors follow random walk process with small covariance. Thus by converting problem into linear estimation problem with known weights, MPF is implemented. Some future work need to be done on modeling weights to get proper performance.

Monte-Carlo simulation results of a well known target tracking problem is considered with two examples. In the first example, sampling interval assumed to be unknown. To estimate the sampling interval, boundaries of the sampling interval are considered as two models in the model set. The weights of the models are estimated using QBAKF, DDMPF and compared with well known AMM, IMM and augmented EKF. The QBAKF performed very well in terms of accuracy and simplicity. The DDMPF performed reasonably well. For the second example, control input matrix is assumed to be unknown. Three different choices of the input-control matrices are considered in the model set. The model weights are estimated using QBAKF, IMM, AMM, and EKF. Again QBAKF performed well in terms of estimating model weights, accuracy and simplicity. This kind of approach can be extended to state estimation for maneuvers by improving modeling part of the weight vectors.

Bibliography

[1] X. R. Li and V. P. Jilkov. Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods. *IEEE Trans. Aerospace and Electronic Systems*, 41(4):1255–1321, Oct. 2005.

[2] J. F. Ru and X. R. Li. Variable-Structure Multiple-Model Approach to Fault Detection, Identification, and Estimation. *IEEE Trans. on Control Systems Technology*, 16(5):1029 – 1038, 2008.

[3] Y. M. Zhang and X. R. Li. Detection and Diagnosis of Sensor and Actuator Failures Using IMM Estimator. *IEEE Trans. Aerospace and Electronic Systems*, AES-34(4):1293–1312, Oct. 1998.

[4] X. R. Li. Hybrid Estimation Techniques. In C. T. Leondes, editor, *Control and Dynamic Systems: Advances in Theory and Applications*, volume 76, pages 213–287. Academic Press, New York, 1996.

[5] Shorten. R, Murray-Smith. R, Bjorgan. R, and Gollee. H. On the interpretation of local models in blended multiple model structures. *International Journal of Control*, 72(7-8):620–628, 1999.

[6] X. R. Li, Zhao. Z, and Li. X. B. General model-set design methods for multiple-model approach. *IEEE Trans. on Automatic Control*, 50(9):1260–1276, 2005.

[7] Narendra. K. S, Driollet. O. A, Feiler. M. A, and George. K. General model-set design methods for multiple-model approach. *Int. J. Adapt. Control and Signal Process.*, 17(2):87–102, 2003.

[8] D. P. Malladi and J. L. Speyer. A New Approach to Multiple Model Adaptive Estimation. In *Proc. 1997 IEEE Conf. on Decision and Control*, pages 3460–3467, San Diego, CA, 1997.

[9] D. P. Malladi and J. L. Speyer. A Generalized Shiryaev Sequential Probability Ratio Test for Change Detection and Isolation. *IEEE Trans. Automatic Control*, AC-44(8):1522–1534, 1999.

[10] M. A. Medina, M. Rodrigues, D. Theilliol, and H. Jamouli. Fault Diagnosis in Nonlinear Systems Through an Adaptive Filter Under a Convex Set Representation. In *Proc. European Control Conference*, 2003.

[11] M. Rodrigues, D. Theilliol, and D. Sauter. Design of An Active Fault Tolerant Control and Polytopic Unknown Input Observer for Systems Described by A Multi-Model Representation. In *Proc. 44th IEEE Conference on Decision and Control and European Control Conference*, volume 44, page 3815, 2005.

[12] M. Verhaegen, R. Hallouzi, and S. Kanev. Fault Detection and Identification using Multiple Models. In *Proc. 7th Portuguese Conference on Automatic Control*, 2006.

[13] Rodrigues. M, Theilliol. D, Adam-Medina. M, and Sauter. D. A fault detection and isolation scheme for industrial systems based on multiple operating models. *Control Engineering Practice*, 16(2):225–239, 2006.

[14] M. Rodrigues, D. Theilliol, M. Adam-Medina, and D. Sauter. A Fault Detection and Isolation Scheme for Industrial Systems Based on Multiple Operating Models. *Control Engineering Practice*, 16(2):225–239, 2008.

[15] R. Hallouzi, M. Verhaegen, and S. Kanev. Multiple Model Estimation: A Convex Model Formulation. *Int. J. Adapt. Control Signal Process.*, 3:217–240, 2009.

[16] S. C. Port. *Theoretical Probability for Applications*. John Wiley & Sons, 1994.

[17] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, 1985.

[18] V. P. Jilkov and X. R. Li. On-Line Bayesian Estimation of Transition Probabilities for Markovian Jump Systems. *IEEE Trans. Signal Processing*, 52(6):1620–1630, June 2004.

[19] A. Doucet, N. de Frietas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[20] T. Schön, F. Gustafsson, and P.J. Nordlund. Marginalized Particle Filters for Nonlinear State-space Models. *IEEE Trans. Signal Processing*, 53(7):22792287, 2005.

[21] D. T. Magill. Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Trans. Automatic Control*, AC-10:434–439, 1965.

[22] H. A. P. Blom. An Efficient Filter for Abruptly Changing Systems. In *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 1984.

[23] Ljung. L. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Trans. on Automatic Control*, 24(1):36–50, 1979.

[24] R. Hallouzi, M. Verhaegen, and S. Kanev. Multiple Model Estimation: A Convex Model Formulation. *Int. J. Adapt. Control and Signal Process.*, 3:217–240, 2008.

[25] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. Wiley, New York, 2001.

[26] Doshi. F and Roy. N. Efficient Model Learning for Dialog Management. In *Proc. of Human Robot Interaction*, 2007.

[27] Doshi. F, Pineau. J, and Roy. N. Reinforcement Learning with Limited Reinforcement: Using Bayes Risk for Active Learning in POMDPs. In *Internaional Conference on Machine Learning*, 2008.

[28] Jaulmes. R, Pineau. J, and Precup. D. Active Learning in Partially Observable Markov Decision Processes. In *European Conference on Machine Learning (ECML)*, 2005.

[29] Jaulmes. R, Pineau. J, and Precup. D. Learning in Non-Stationary Partially Observable Markov Decision Processes. In *ECML Workshop on Reinforcement Learning in Non-Stationary Environments*, 2005.

[30] Papoulis. A. *Probability, Random variables and Stochastic processes*. McGraw-Hill, 1991.

[31] D. Kazakos. Recursive Estimation of Prior Probabilities Using a Mixture. *IEEE. Trans. Information Theory*, IT-23:203–211, March 1977.

[32] G. R. Dattatreya and L. N. Kanal. Estimation of Mixing Probabilities in Multiclass Finite Mixtures. *IEEE Trans. Systems, Man, and Cybernetics*, SMC-20(1):149–158, Jan./Feb. 1990.

[33] J.-L. Gauvain and C.-H. Lee. Maximum a Posteriori Estimation for Multivariate Mixture Observations of Markov Chains. *IEEE Trans. Speech Audio Processing*, 2(2):291–298, April 1994.

[34] Q. Huo, C. Chan, and C.-H. Lee. Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition. *IEEE Trans. Speech Audio Processing*, 3(5):334–345, Sept. 1995.

[35] Q. Huo and C.-H. Lee. On-Line Adaptive Learning of the Continuous Density Hidden Markov Model Based on Approximate Recursive Bayes Estimate. *IEEE Trans. Speech Audio Processing*, 5(2):161–172, March 1997.

[36] C.-H. Lee and Q. Huo. On Adaptive Decision Rules and Decision Parameter Adaptation for Automatic Speech Recognition. *Proceedings of the IEEE*, 88(8):1241–1269, Aug. 2000.

[37] R. Hallouzi, M. Verhaegen, R. Babuska, and S. Kanev. Model weight and state estimation formultiplemodel systems applied to fault detection and identification. In *In Proceedings of the IFAC Symposium on System Identification*, 2006.

[38] A. Doucet and C. Andrieu. Iterative Algorithms for State Estimation of Jump Markov Linear Systems. *IEEE Trans. Signal Processing*, 49(6):1216–1227, June 2001.

[39] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995.

[40] X. R. Li and Z.-L. Zhao. Evaluation of Estimation Algorithms—Part I: Incomprehensive Performance Measures. *IEEE Trans. Aerospace and Electronic Systems*, AES-42(4):1340–1358, Oct. 2006.

Vita

The author was born in 1985, in the city of Nalgonda, India. Mr. Katkuri completed his bachelors in Electronics and Communication Engineering from Osmania University, Hyderabad India in 2006 with distinction. He worked as Senior Design Engineer at MSDPL (Locomotive control systems based company), Hyderabad India from 2006-2008. He finished his Masters in Electrical Engineering from the University Of New Orleans in Jun'2010 with a cumulative GPA of 3.80. The author worked with Dr. V. P. Jilkov as a Research Assistant in his Masters.

The author can be reached at `http://ece.engr.uno.edu/isl/People.htm`.