Ensemble Kalman Estimation with Gaussian Mixture Models

by:

Ruoxia Li

A thesis submitted
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

The objective of this work is to study the problems that arise in state estimation for severely nonlinear systems. In practice, many processes are nonlinear, accompanied by uncertain parameters. The complexity of the model causes the probability density function (PDF) of the states to deviate from a Gaussian distribution. This presents a challenge for Kalman-based state estimators such as the extended Kalman filter, since they model the state PDF as Gaussian. In order to achieve more accurate estimation, the modeling of the state distribution needs to be improved. The first problem is to develop an estimator for the state PDF of arbitrary distribution. In this work, we develop an estimator based on a Gaussian mixture model (GMM) coupled with the ensemble Kalman filter (EnKF) specifically for estimation with multimodal state distributions.

The second problem is that the conventional recursive state estimation procedures cannot handle inequality constraints on the states. Therefore, they might result in physically meaningless or non-convergent estimates, especially when the initial values are poor. The incorporation of constraints can help improve the estimation significantly. In this work, we develop a novel state estimation technique to incorporate inequality constraints for the case of Gaussian filters. Furthermore, we consider the constrained estimation for the case where the state PDF cannot be approximated with a Gaussian distribution. To this end, we develop a framework to incorporate the inequality constraints for the GMM based EnKF mentioned in the first problem.

The filtering provides the estimates by assimilating the history data. Such estimation can be further improved by using the smoothing technique which assimilate all the available data. Therefore the third problem is to develop a smoothing framework for the systems whose state PDF is non-Gaussian. In this work, we extend the existing ensemble Kalman smoother (EnKS) to deal with non-Gaussian systems by combining it with the GMM model.

The thesis focuses on three aspects in data assimilation problem. The first is filtering problem for non-Gaussian systems. The second is smoothing problem for non-Gaussian systems. The third is the incorporation of inequality constraints into estimation. The thesis provides the theoretical deduction of our proposed approaches as well as practical applications.

Preface

This thesis is an original work by Ruoxia Li. The materials presented in this work are under the supervision of Dr. Biao Huang and Dr. Vinay Prasad, and are funded by Natural Sciences and Engineering Research Council of Canada (NSERC) and China Scholarship Council (CSC).

Chapter 2 of the thesis is based on the paper published as R. Li, V. Prasad, and B. Huang, "Gaussian Mixture Model-Based Ensemble Kalman Filtering for State and Parameter Estimation for a PMMA Process", *Processes*, 4(2), 9, doi: 10.3390/pr4020009, 2016. Part of chapter 3 is based on the paper accepted by European Control Conference (2018). I was responsible for the theoretical study and algorithm development. Dr. Biao Huang and Dr. Vinay Prasad were supervisory authors and were involved in the manuscript editing.

Part of chapter 3 was accepted for publication by European Control Conference (2018). The rest of chapter 3 is under preparation for publication for Journal of Process Control. Chapter 4 is under preparation for publication. For chapters 3 and 4, I was responsible for the theoretical development, algorithm design and manuscript composition. Dr. Nabil Magbool Jan participated in the design of the optimization algorithm. Dr. Biao Huang and Dr. Vinay Prasad supervised the work and were involved in the manuscript editing.

Chapter 5 is under preparation for publication. I was responsible for the theoretical development, algorithm design and manuscript composition. Dr. Biao Huang and Dr. Vinay Prasad supervised the work and were involved in the manuscript editing.

Acknowledgement

I would like to express my sincere gratitude to my supervisors Dr. Biao Huang and Dr. Vinay Prasad for their inspiration, patience and support throughout my doctorate study. I thank Dr. Huang for always pointing out the big picture view of research for me and providing the excellent academic resources for me to fulfill my purpose. I thank Dr. Prasad for his patience to discuss every detail of my work and all the help he provided whenever I encountered difficulties in research. I thank both of them for all the valuable critiques I received for my work and the guidance not only in my study but also in life.

I would like to thank the Chinese Scholarship Council (CSC) and National Science and Engineering Research Council (NSERC) of Canada for the financial support. I also thank the Department of Chemical and Materials Engineering of University of Alberta for giving me the opportunity to pursue my doctorate degree. I thank the lovely staff who once helped me at some point of my doctorate study.

I would also like to thank of the members of the CPC group, especially Dr. Nabil Magbool Jan for helping me with his expertise in optimization area. I have made productive progress ever since I started to work with him. He has also been of great help with editing my publications and thesis. I would like to thank Dr. Shunyi Zhao and Dr. Yuan Yuan for all the knowledge they provided during our discussions. I also thank the other group members for their direct or indirect contributions to my thesis. I learn a lot in every well-prepared presentation they provided in the group meetings. I also enjoy the company and friendship that made my campus life always joyful and colorful.

Last but not least, I express my deepest thanks to my parents for their nurture, love and support. I gained strength from them every time I felt down. I also thank my husband Dr. Ning He for his company and encouragement at every step of my doctorate study. Without them, I can never go through this arduous journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

The problem of state estimation arises when we try to understand the inside information of a black-box system that is too costly or impossible to measure physically. The acquirement of this hidden information, referred to as the states, is crucial for engineers to better control and maintain the systems. The formulation of the Bayes' theory is suitable for deriving the states information based upon the measurements available from the system. Over the years, the Bayes' theory has been successfully applied to develop the stochastic online state estimators. Among the Bayesian estimators, the Kalman filter is definitely the milestone of state estimation area. It is the optimal filter for linear systems with Gaussian noises, however, its performance can be severely compromised for the reasons including:
(1) Severe nonlinearity of the model equations;
(2) High dimensionality of the state space;
(3) Non-Gaussian shape of the state distribution.

The weakness of the Kalman filter leads to the development of the nonlinear Bayesian estimation algorithms to provide the most appropriate estimator for systems with specific requirements. If the nonlinearity is our primary concern, the extended Kalman filter which is based on the first-order Taylor series expansion is the most straightforward resort. If the nonlinearity is so severe that the first-order Taylor expansion is not a sufficient approximation, the unscented Kalman filter could be a better option. The ensemble Kalman filter can deal with system that is not only nonlinear, but also high dimensional. All the three filters mentioned are based on the assumption that the state distribution is assumed Gaussian. If we relax the limit of the system even more to non-Gaussian state distribution, the particle filter so far is the best option. Although the particle filter can give all the values of the moments of the posterior distribution, it cannot provide an analytic expression for posterior probability density function (PDF). One of the motivations of this thesis is to provide an estimator developed under the Bayesian framework for nonlinear and non-Gaussian systems. This estimator is expected to overcome some of the inherent shortcomings of the particle filter, such as non-robust and degeneracy problem.

In state estimation problems, the lack of knowledge of the model parameters or the operating conditions could result in estimation which might be physically meaningless, such as negative flow rate etc. Besides many real processes might have bounds on some of the states. The incorporation of the constraints on the states can help bring the estimation closer to the true value and eliminate the unfeasible estimation. Two issues need to be taken into consideration for constrained state estimation problem. One is the formulation of the constraints. Most constraints can be formulated as an equality or inequality constraints, which is the focus of the existing work. In some cases, the constraints are nonlinear. The second is how to enforce the constraints, especially how to integrate it into the Bayesian estimation framework. The most intuitive technique of implementing the constraints is through the clipping method by simply getting rid of the part of the unconstrained solution which violates the constraints. A more sophisticated approach is to use the optimization-based method to include the linear or nonlinear constraints. The design of optimization has to be carefully designed for different systems and estimators. Another motivation of this thesis is to develop a constrained state estimation technique for the nonlinear and non-

Gaussian systems.

1.2 Data assimilation

1.2.1 Recursive Bayesian estimation

The goal of data assimilation is to track the performance of a physical process using the its measurable outputs. Two elements are vital in data assimilation: a model to describe the dynamics of the process as well as observations. The state space model is the most commonly used representation of the true physical process. The transition of the states x in the state space model is described as follows:

$$\frac{dx}{dt} = f(x, t) \tag{1.1}$$

Eq.(1.1) describes the evolution of the states of a deterministic system modeled by with function $f(x, t)$. In reality, the process is influenced by so many factors that $f(x, t)$ can not include them all. Therefore a stochastic model is a better approximation to account for the discrepancy between the true state evolution and the model prediction. Such stochastic model is given by:

$$\frac{dx}{dt} = f(x, t) + w(x, t) \tag{1.2}$$

where $w(x, t)$ is usually referred to as system noise or model error.

Besides the state transition function $f(x, t)$, the state space model also includes a measurement or observation model. We have its stochastic formulation given by:

$$y(t) = h(x, t) + v(x, t) \tag{1.3}$$

This measurement model h(.) builds a relationship between the state space and the measurement space, which enables the observations to correct the state in return when it diverges from the true trajectory.

The stochastic feature of the state space model leads to state estimation with uncertainty which can only be described using the probabilistic formulation instead of a point value.

Definition 1.1(Probabilistic formulation of the state space model)
Assuming the states follow the Markov model, a stochastic state space model can be formulated in the following way:

$$\begin{aligned} x_0 &\sim p(x_0) \\ x_k &\sim p(x_k|x_{k-1}) \\ y_k &\sim p(y_k|x_k) \end{aligned}$$

The purpose of the state estimation is to obtain the full joint distribution $p(x_{0:k}|y_{1:k})$, where $x_{0:k} = \{x_0, \dots, x_k\}$ and $y_{1:k} = \{y_1, \dots, y_k\}$. With a straightforward Bayes' rule, we get:

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_{1:k}|x_{0:k})p(x_{0:k}|y_{1:k-1})}{p(y_{1:k})} \tag{1.4}$$

The biggest disadvantage of this joint distribution is the computation increases dramatically with the time steps since the dimensionality increases with the time steps. Eventually it will become computationally intractable. Therefore a recursive Bayes' framework to only compute the marginal distribution $p(x_k|y_{1:k})$ online. Apply the Bayes' rule on $p(x_k|y_{1:k})$, we get:

$$\begin{aligned} p(x_k|y_{1:k}) &= \frac{p(y_k|x_{1:k}, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} && (1.5) \\ &\propto p(y_k|x_k)p(x_k|y_{1:k-1}) && (1.6) \end{aligned}$$

The denominator $p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k$ is a constant, denoted by C_k.

The following two properties have been applied to derive from Eq.(1.5) to Eq.(1.6).

Markov Property of the states:

By assuming the state sequence follows the Markov property, it means that the state at current time step x_k only depends on the state at the previous time step and is independent of the states and measurements before $k-1$. Furthermore, x_k is also independent of the states and measurements beyond $k+1$.

$$p(x_k|x_{1:k-1}, y_{1:k-1}) = p(x_k|x_{k-1})$$

$$p(x_k|x_{k+1:T}, y_{k+1:T}) = p(x_k|x_{k+1})$$

Independence of measurements:

The other assumption is that the measurement at current time step y_k is conditionally independent of the states and measurements at all previous steps:

$$p(y_k|x_{1:k}, y_{1:k-1}) = p(y_k|x_k) \tag{1.7}$$

It has to be noted that Eq.(1.6) is called a Bayesian filter. The general state estimation problem can be classified into the following three categories.

Definition 1.2:(Filter, smoother and predictor)

For a posterior distribution $p(x_k|y_{1:T})$ obtained through Bayes' rule:

- If $k < T$, the estimator is called a smoother, which uses the future measurements to update the current states.
- If $k = T$, the estimator is called a filter, which uses all the measurements up to the current time step to update the current states.
- If $k > T$, the estimator is called a predictor, which predicts the future states using all the measurements up to the current time step.

Eq.(1.6) explains how the Bayesian inference is performed for a filter and also presents the recursive structure of Bayesian filter. The term $p(x_k|y_{1:k-1})$ is called a predicted distribution of the state at time step k and $p(y_k|x_k)$ is called a likelihood term to update the predicted distribution given the measurements at time step k. In conclusion, we have the recursive estimation algorithms performed in the following description.

Recursive Bayesian filter:

- *Initialization:* When k=0, the filter starts from a prior distribution $p(x_0)$.
- *Prediction:* The filter calculates the predicted distribution $p(x_k|y_{1:k-1})$ using the following Chapman-Kolmogorov equation [21]:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{x-1} \tag{1.8}$$

- *Update:* The filter updates the predicted distribution by incorporating the latest measurement using the likelihood distribution.

$$p(x_k|y_{1:k}) = \frac{1}{C_k}p(x_k|y_{1:k-1})p(y_k|x_k) \tag{1.9}$$

With $p(x_k|y_{1:k})$ obtained, the moments of the states can be calculated easily. Furthermore this posterior distribution serves as the prior distribution for the loop of the time step $k+1$.

The smoother, as is explained in definition 1.2, performs the state estimation by incorporating the future observations. The smoother can either pursue a joint posterior distribution $p(x_{0:T}|y_{1:T})$ or a marginal posterior distribution $p(x_k|y_{1:T})$, depending on the specific formulation of the smoother. Whichever one a smoother goes for, it can be categorized as either a fix-interval smoother or a fix-lag smoother or a fix-point smoother [88].

Definition 1.3: (Fixed-interval, fixed-lag and fixed-point smoothers)

- Fixed-interval smoother estimates the state, x_k^a, at every time point of a given interval $[0 \quad T]$, given the measurements from 1 to T, $y_k, k = 1, \ldots, T$.
- Fixed-lag interval smoother estimates the state at time step T-L, x_{T-L}^a, given the

measurements from T-L to T, $y_k, T - L < k < T$.

- Fixed-point smoother estimates the state at one specific time step k, x_k^a, give the measurements from 1 to T, $y_k, k = 1, \ldots, T$.

In the following, we introduce the general formulation for backward Bayesian smoother which targets the marginal posterior distribution. The smoothing distribution of x_k is calculated by marginalizing the joint distribution of x_k and x_{k+1}.

$$p(x_k|y_{1:T}) = \int p(x_k|x_{k+1}, y_{1:T})p(x_{k+1}|y_{1:T})dx_{k+1} \tag{1.10}$$

Using the Markov property:

$$\begin{aligned} p(x_k|x_{k+1}, y_{1:T}) &= p(x_k|x_{k+1}, y_{1:k}) && (1.11) \\ &= \frac{p(x_k, x_{k+1}|y_{1:k})}{p(x_{k+1}|y_{1:k})} && (1.12) \\ &= \frac{p(x_{k+1}|x_k)p(x_k|y_{1:k})}{p(x_{k+1}|y_{1:k})} && (1.13) \end{aligned}$$

Substituting Eq.(1.13) into Eq.(1.10), we have:

$$p(x_k|y_{1:T}) = p(x_k|y_{1:k}) \int \frac{p(x_{k+1}|x_k)p(x_{k+1}|y_{1:T})}{p(x_{k+1}|y_{1:k})} dx_{k+1} \tag{1.14}$$

1.2.2 Linear Bayesian filter and smoother

1.2.2.1 Kalman filter

The famous Kalman filter [47], named after its proposer Rudolph E. Kalman, is the optimal Bayesian filter for linear systems. Its remarkably simple formulation for data formulation lays the foundation of this work. Numerous studies have been published ever since its introduction. For a systematic tutorial of Kalman filter, refer to [64][59].

Although Eq.(1.8) and Eq.(1.9) give the optimal solution for the recursive Bayesian estimation, the solution usually does not have a closed-form expression except when several strict conditions are applied. The Kalman filter is essentially the closed-form solution when the model satisfies the following conditions:

1. The model $f(x,t)$ is linear. The discrete linear state space model is written as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1.15}$$

$$y_k = Hx_k + v_k \tag{1.16}$$

2. The process noise v_k and measurement noise w_k are both Gaussian stationary white noise sequences which have the following properties:

$$E(w_k) = 0 \tag{1.17}$$

$$E(v_k) = 0 \tag{1.18}$$

$$E(w_k w_j^T) = Q \quad when \;\; k = j \qquad E(w_k w_j^T) = 0 \quad when \;\; k \neq j \tag{1.19}$$

$$E(v_k v_j^T) = R \quad when \;\; k = j \qquad E(v_k v_j^T) = 0 \quad when \;\; k \neq j \tag{1.20}$$

$$E(v_k w_j^T) = 0 \tag{1.21}$$

Moreover, the states and process and measurement noises are uncorrelated:

$$E(x_k v_k^T) = 0 \tag{1.22}$$

$$E(x_k w_k^T) = 0 \tag{1.23}$$

With the assumptions above, Eq.(1.8) can be written as:

$$\begin{aligned} p(x_k|y_{1:k-1}) &= \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \\ &= \int N(x_k|Ax_{k-1}, Q_{k-1})N(x_{k-1}|\mu_{k-1}, P_{k-1})dx_{k-1} \\ &= N(x_k|A\mu_{k-1}, AP_{k-1}A^T + Q_{k-1}) \end{aligned} \tag{1.24}$$

where the notation *N(.)* represents a Gaussian distribution and μ_{k-1} and P_{k-1} represent the mean and covariance of previous time step. Denote $\mu_k^f = A\mu_{k-1}$ and $P_k^f = AP_{k-1}A^T + Q_{k-1}$, where μ_k^f and P_k^f are the predicted mean and covariance of time step k.

The joint distribution of x_k and y_k is given by:

$$\begin{aligned} p(x_k, y_k | y_{1:k-1}) &= p(y_k|x_k)p(x_k|y_{1:k-1}) \\ &= N(\begin{pmatrix} x_k \\ y_k \end{pmatrix} |\mu_1, P_1) \end{aligned} \tag{1.25}$$

where

$$\mu_1 = \begin{pmatrix} \mu_k^f \\ H\mu_k^f \end{pmatrix} \qquad P_1 = \begin{pmatrix} P_k^f & P_k^f H^T \\ HP_k^f & HP_k^f H^T + R_k \end{pmatrix} \tag{1.26}$$

Integrating the joint distribution over y_k, we have the marginal distribution of the x_k given by:

$$p(x_k|y_{1:k}) = N(x|\mu_k^a, P_k^a) \tag{1.27}$$

where

$$\mu_k^a = \mu_k^f + P_k^f H^T (HP_k^f H_k^T + R_k)^{-1}(y_k - H\mu_k^f) \tag{1.28}$$

$$P_k^a = P_k^f - P_k^f H^T (HP_k^f H_k^T + R_k)^{-1} HP_k^f \tag{1.29}$$

μ_k^a is the updated mean and the point estimate we aim to obtain. P_k^a is the error covariance of the estimate. The common term in Eq.(1.28) and Eq.(1.29), $P_k^f H^T (HP_k^f H_k^T + R_k)^{-1}$, is defined to be the well-known Kalman gain:

$$K = P_k^f H^T (HP_k^f H_k^T + R_k)^{-1} \tag{1.30}$$

The derivation above starts from the general recursive Bayesian estimation formulation and arrives at the closed-from expression of the updated mean and covariance as well as the Kalman gain from the Bayesian inference perspective. In the following, we show how to achieve the same results from the optimization angle, which explains the optimality of the Kalman filter.

Let the state estimation at the current time step k be the summation of two parts: a prediction part x_k^f given by the open loop prediction of the model and a update part which linearly updates the prediction using the the latest observation y_k:

$$x_k^f = Ax_{k-1}^a + Bu_{k-1} \tag{1.31}$$

$$x_k^a = x_k^f + K(y_k - Hx_k^f) \tag{1.32}$$

The error covariance of the prediction step P_k^f can be calculated by propagating the prior covariance P_{k-1}^a through the model:

$$P_k^f = AP_{k-1}^a A^T + Q_{k-1} \tag{1.33}$$

The estimation error at the prediction step and the update step are given by:

$$\epsilon_k^f = x_k - x_k^f \tag{1.34}$$

$$\epsilon_k^a = x_k - x_k^a \tag{1.35}$$

where x_k is the true state value.

The goal is to find the gain K such that x_k^a produces the smallest estimation error covariance. The following optimization problem is formulated to achieve this purpose:

$$K = arg \min_K P_k^a = arg \min_K E(\epsilon_k^a \epsilon_k^{aT}) \tag{1.36}$$

Since the predicted covariance is already known, we need to write ϵ_k^a in terms of ϵ_k^f. With a slight manipulation of Eq.(1.34) and Eq.(1.35), by substituting them into Eq.(1.32), we have:

$$\epsilon_k^a = \epsilon_k^f - K(H\epsilon_k^f + w_k) \tag{1.37}$$

Then the updated error covariance can be written as:

$$\begin{aligned} P_k^a &= E(\epsilon_k^a \epsilon_k^{aT}) \\ &= E((\epsilon_k^f - K(H\epsilon_k^f + w_k))(\epsilon_k^f - K(H\epsilon_k^f + w_k))^T) \\ &= P_k^f - P_k^f H^T K + KRK^T - KHP_k^f + KHP_k^f H^T K^T \end{aligned} \tag{1.38}$$

Optimizing Eq.(1.38) over K, we have the Kalman gain given by:

$$K = P_k^f H^T (HP_k^f H^T + R)^{-1} \tag{1.39}$$

Substituting Eq.(1.39) into Eq.(1.38), we have the updated covariance given by:

$$P_k^a = P_k^f - P_k^f KH \tag{1.40}$$

The estimated result given by Eq.(1.32) and Eq.(1.40) using the Kalman gain K given by Eq.(1.39) is the minimal error variance state estimation, hence the name optimal filter. It is consistent with the estimation result obtained from the Bayesian inference given by Eq.(1.28) and Eq.(1.29), so is the formulation of the Kalman gain.

1.2.2.2 Rauch-Tung-Striebel smoother

Various linear optimal smoothers have been proposed for achieving different estimation goals, such as the fixed-interval smoother as proposed in [2], the fixed-lag smoother smoother in

[68], the fixed-point smoother in [9], the RauchTungStriebel (RTS) smoother in [78] and two-filter smoother in [27]. However, they are all Kalman update based and are essentially equivalent with each other as long as the system is linear and the Gaussian state distribution assumption stands.

Out of the aforementioned smoothers, the RTS smoother is the most popular and widely used one in the literature. It can be formulated as a fix-interval smoother and can be modified to perform the fix-lag smoother as well. The RTS smoother consists of two runs, a forward run completed by the Kalman filter to obtain the posterior distribution and a backward run to further update the posterior distribution. If we assume the state distribution is $p(x_k|y_{1:k}) = N(x_k|x_k^a, P_k^a)$ for every time step of the forward filtering, the backward recursions to calculate the smoothed mean x_k^s and covariance P_k^s are given as follows:

$$x_k^s = x_k^a + G(x_{k+1}^s - x_{k+1}^f) \tag{1.41}$$

$$P_k^s = P_k^a + G(P_{k+1}^s - P_{k+1}^f)G^T \tag{1.42}$$

where G is the smoothing gain and is given by:

$$G = P_k^a A^T [P_{k+1}^f]^{-1} \tag{1.43}$$

x_{k+1}^f and P_{k+1}^f are the predicted mean and covariance of k+1.

$$x_{k+1}^f = Ax_k^a \tag{1.44}$$

$$P_{k+1}^f = AP_k^a A^T + Q \tag{1.45}$$

One appealing feature of the RTS smoother is that the backward recursions do not need any more forward model runs if the predicted mean and covariance are stored during the forward filtering step.

1.2.3 Nonlinear Bayesian filter and smoother

Over the years, numerous work has been done on the state estimation for nonlinear systems. Essentially, all of them attempt to deal with the nonlinearity by using different approximation approaches, therefore they only can provide sub-optimal estimation. In this section, we review some of the most widely used nonlinear filters and smoothers. The general form of a discrete nonlinear system is given by:

$$x_{k+1} = f(x_k, u_k) + w_k \tag{1.46}$$

$$y_{k+1} = h(x_{k+1}) + v_{k+1} \tag{1.47}$$

1.2.3.1 Extended Kalman filter and smoother

The extended Kalman filter (EKF) [43][64] is the most straightforward extension of the Kalman filter. The prediction step can be carried out without difficulty for nonlinear systems:

$$x^f_{k+1} = f(x^a_k) + w_k$$

However, the propagation of the covariance given in Eq.(1.33) does not stand any more because of the nonlinearity of the system. Therefore the most intuitive approach is to simply linearize the nonlinear dynamics so that the Kalman filter can be applied, which is the method that EKF resorts in order to handle nonlinearity. The EKF linearizes the process model by using the Taylor expansion, and Eq.(1.46) becomes:

$$\begin{aligned} x_{k+1} &= f(x_k) + w_k \\ &= f(x^a_k) + J_f(x^a_k)(x_k - x^a_k) + \cdots + w_k \end{aligned} \tag{1.48}$$

where J_f is the Jacobian matrix given by:

$$J_f = \frac{\partial f}{\partial x}|_{x_k^a, u_k} \tag{1.49}$$

Ignoring the higher order terms, Eq.(1.48) becomes:

$$\begin{aligned} x_{k+1} &\approx f(x_k^a) + J_f(x_k - x_k^a) + w_k \\ x_{k+1} - x_{k+1}^f &\approx J_f(x_k - x_k^a) + w_k \\ e_{k+1}^f &\approx J_f e_k^a + w_k \end{aligned} \tag{1.50}$$

Therefore the predicted covariance is given by:

$$P_{k+1}^f = J_f P_k^a J_f^T + Q_k \tag{1.51}$$

For the measurement model, apply the Taylor approximation again, and linear operator of h(.) is given by:

$$J_h = \frac{\partial h}{\partial x}|_{x_{k+1}^f} \tag{1.52}$$

$$\begin{aligned} y_k &\approx h(x_{k+1}^f) + J_h(x_{k+1}^f)(x_{k+1} - x_{k+1}^f) + v_{k+1} \\ y_k - h(x_{k+1}^f) &= J_h(x_{k+1}^f) e_{k+1}^f + v_{k+1} \end{aligned} \tag{1.53}$$

Repeating the optimization problem of Eq.(1.36) with the approximation given by Eq.(1.50) and Eq.(1.53), we have the Kalman gain given by:

$$K = P_k^f J_h^T (J_h P_k^f J_h^T + R)^{-1} \tag{1.54}$$

The updated covariance is given by:

$$P_{k+1}^{a} = (I - KJ_h)P_{k+1}^{f} \tag{1.55}$$

From the composition of the Kalman gain, we can see that the EKF basically replaces linear matrix A and H in the linear model with the approximated linear operators J_f and J_H. In this section, the principle of the EKF is reviewed. In chapter three, the EKF algorithm is described again in a more concise fashion.

In the following, we apply the same linearization idea to the Kalman smoother described in Eq.(1.41) and Eq.(1.42) to formulate the RTS Kalman extended smoother[74].

$$\begin{aligned} x_k^s &= x_k^a + G(x_{k+1}^s - x_{k+1}^f) \\ P_k^s &= P_k^a + G(P_{k+1}^s - P_{k+1}^f)G^T \end{aligned}$$

where

$$x_{k+1}^f = f(x_k^a) \tag{1.56}$$

$$P_{k+1}^f = J_f P_k^a J_f^T + Q \tag{1.57}$$

The smoother gain G is given by:

$$G = P_k^a J_f^T (J_f P_k^a J_f^T + Q)^{-1} \tag{1.58}$$

J_f is exactly the same as in the EKF, i.e.:

$$J_f = \frac{\partial f}{\partial x}|_{x_k^a, u_k} \tag{1.59}$$

1.2.3.2 Unscented Kalman filter and smoother

The unscented Kalman filter (UKF) [101] is based on theory of the unscented transformation (UT) which calculates the statistics of a random variable undergoing nonlinear transformation [45]. Specifically a fixed number of weighted deterministic points, which are called the sigma points, are selected to represent the distribution of the original random variable. Each sigma point can be considered as a realization of the random variable. Therefore the stochastic information can be derived from the weighted sigma points. The sigma points are selected using the following procedure.

Considering a random variable $x \in R^n$, $x \sim N(x, \mu, P)$,a matrix χ which consists of the (2n+1) sigma points are given by:

$$\begin{aligned}
&\chi^0 = \mu \\
&\chi^i = \mu + \sqrt{(n+\lambda)P}, \quad i = 1, 2, \ldots, n \\
&\chi^i = \mu - \sqrt{(n+\lambda)P}, \quad i = n+1, n+2, \ldots, 2n
\end{aligned}$$

where λ is a scaling parameter. The weights associated with each sigma points are given by:

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{n+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{n+\lambda} \\
W_i^{(m)} &= \frac{1}{2(n+\lambda)}, \quad i = 1, 2, \ldots, 2n \\
W_i^{(c)} &= \frac{1}{2(n+\lambda)}, \quad i = 1, 2, \ldots, 2n
\end{aligned}$$

Same as the Kalman filter, the UKF also attempts to estimate the mean and covariance of the posterior distribution using the Kalman update. The key difference is that the UKF calculates the mean and covariance using the UT instead of explicitly propagating and updating them as in the Kalman filter. The UKF algorithm is described as follows:

Prediction:

1. Propagate the sigma points through the model:

$$\chi_{k+1}^{i} = f(\chi_k^i) + w_k^i, \quad i = 0, \ldots, 2n \tag{1.60}$$

2. Calculate the predicted mean and covariance from $\chi_{k+1}^i, i = 0, \ldots, 2n$ using:

$$x_{k+1}^{f} = \sum_{i=0}^{2n} W_i^{(m)} \chi_{k+1}^i \tag{1.61}$$

$$P_{k+1}^{f} = \sum_{i=0}^{2n} W_i^{(c)} (\chi_{k+1}^i - x_{k+1}^f)(\chi_{k+1}^i - x_{k+1}^f)^T + Q \tag{1.62}$$

Update:

3. Calculate the predicted observations corresponding to each sigma point:

$$y_{k+1}^{i} = h(x_{k+1}^i) + v_{k+1}^i \tag{1.63}$$

4. Calculate the mean and the covariance of the observation space and the cross covariance of the state and the observation:

$$y_{k+1}^{f} = \sum_{i=0}^{2n} W_i^{(m)} y_{k+1}^i, \quad i = 0, \ldots, 2n \tag{1.64}$$

$$P_{k+1}^{yy} = \sum_{i=0}^{2n} W_i^{(c)} (y_{k+1}^i - y_{k+1}^f)(y_{k+1}^i - y_{k+1}^f)^T + R \tag{1.65}$$

$$P_{k+1}^{xy} = \sum_{i=0}^{2n} W_i^{(c)} (\chi_{k+1}^i - x_{k+1}^f)(y_{k+1}^i - y_{k+1}^f)^T \tag{1.66}$$

5. Calculate the Kalman gain and the updated mean and covariance:

$$K = P^{xy}_{k+1}[P^{yy}_{k+1}]^{-1} \tag{1.67}$$

$$x^a_{k+1} = x^f_{k+1} + K(y_{k+1} - y^f_{k+1}) \tag{1.68}$$

$$P^a_{k+1} = P^f_{k+1} - KP^{yy}_{k+1}K^T \tag{1.69}$$

The UKS works very similar to the Kalman filter except that it calculates the covariance matrices for computation of the Kalman gain using the weighted sigma points. Applying this very same idea to the RTS Kalman smoother framework, we have backward smoothing of the unscented RTS smoother given as follows [82].

1. Store the predicted sigma points χ^i_{k+1} and the updated sigma points χ^i_k of every step of the filtering.

2. Calculate the auto-covariance of time step k+1 and cross-covariance of both time steps k and k+1.

$$S_{k+1} = \sum_{i=0}^{2n} W^{(c)}_i (\chi^i_{k+1} - x^f_{k+1})(\chi^i_{k+1} - x^f_{k+1})^T \tag{1.70}$$

$$C_{k+1} = \sum_{i=0}^{2n} W^{(c)}_i (\chi^i_k - x^a_k)(\chi^i_{k+1} - x^f_{k+1})^T \tag{1.71}$$

$$G = C_{k+1}[S_{k+1}]^{-1} \tag{1.72}$$

$$x^s_k = x^a_k + G(x^s_{k+1} - x^f_{k+1}) \tag{1.73}$$

$$P^s_k = P^a_k + G(P^s_{k+1} - P^f_{k+1})G^T \tag{1.74}$$

1.2.3.3 Ensemble Kalman filter and smoother

The study of the ensemble Kalman filter (EnKF) and ensemble Kalman smoother (EnKS) and their extensions will be the focus of this thesis. In the following chapters, mathematical details of these two algorithms will be described. In this section, we will discuss the philosophy behind this Monte Carlo method based data assimilation approach.

The EnKF deals with the nonlinearity using the same idea of the UKF-calculating the statistical information of the state space through an ensemble of samples of the state space, except that the generation of the samples are different between the two approaches. The UKS uses the sigma points obtained through the UT to represent the state space, while the EnKF resorts to the Monte Carlo sampling technique. Therefore the samples of the UKS are deterministic, but stochastic for the EnKF. At the initial time step, the EnKF produces an ensemble of particles $\{x_0^i\}_{i=1,\ldots,N}$ from the initial distribution $p(x_0)$ which usually comes from prior knowledge. Then each particle of this ensemble is propagated through the model and updated using the Kalman update formulation at every time step. The evolution of the ensemble shapes the predicted and updated distribution. The predicted and updated mean and covariance as well as the Kalman gain are all calculated using the ensemble, which is the same as in the UKF. One prominent difference between the EnKF and the UKF is that the ensemble members are equally weighted. Figure 1.1 describes the principle of the EnKF.

Another difference between the two filters is the number of the samples required by the filtering. The number of the sigma points is fixed to be $2n + 1$, where n is the dimensionality of the state space. Therefore this number grows linearly with the dimensionality of the state space, which is not necessarily the same for the EnKF. Numerous studies have proven that even with high-dimension systems, the EnKF can still deliver a fair estimation with small number of sample points, which is the advantage of the EnKF over the UKF.

The RTS EnKS is formulated similarly to the UKS, except that the weight terms in S_{k+1} and C_{k+1} in Eq.(1.70) and Eq.(1.71) have to be removed since the ensemble members are equally weighted. Note that the initial EnKS proposed by Evensen et al.[25] is a fix-lag smoother. It does not employ the RTS smoother framework. We will introduce the details of both smoothers in chapter 5.

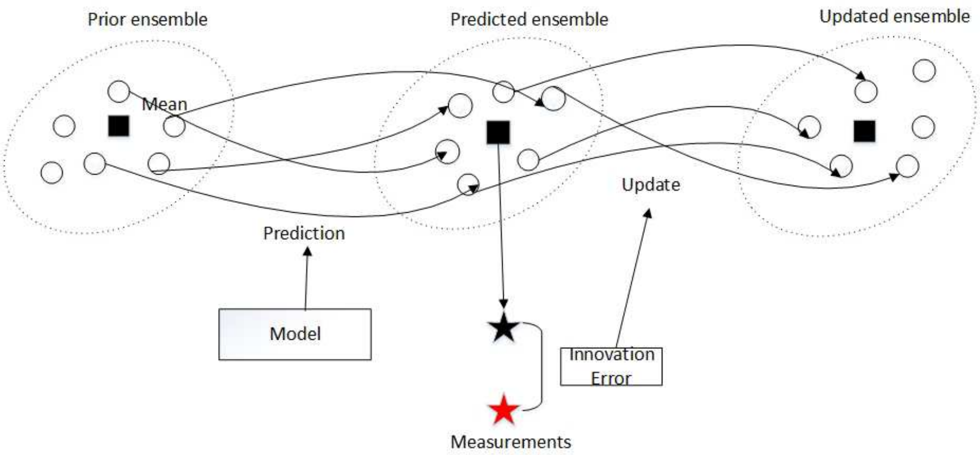


Figure 1.1: The working principle of the EnKF

1.2.3.4 Particle filter and smoother

The particle filter (PF) [16][3] has received great attention in recent years. Unlike the filters introduced previously, the PF is not based on the Kalman update. That is to say it does not use the linear correction to assimilate the observations. Moreover, it is not limited to the assumption of Gaussian prior and posterior distribution because it estimates the full posterior distribution instead of the first two moments.

The PF is a sequential Monte Carlo (SMC) method, which utilizes random samples, referred to as particles, generated using the Monte Carlo method to represent the state distribution. Assuming N particles are used for the prior distribution $p(x_k|y_k)$ of time step k+1, the Monte Carlo approximation of the $p(x_k|y_k)$ can be written as:

$$p(x_k|y_k) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_k^i) \tag{1.75}$$

Since $p(x_k|y_k)$ is arbitrary, it is difficult to sample directly from it, therefore the concept of importance sampling is introduced to sample from an easier distribution $q(x_k|y_k)$, referred to as the importance density. Now the importance weights are defined as:

$$w(x_k) = \frac{p(x_k|y_k)}{q(x_k|y_k)} \tag{1.76}$$

Then $p(x_k|y_k)$ can be written as:

$$\begin{aligned} p(x_k|y_k) &= w(x_k)q(x_k|y_k) \\ &= \sum_{i=1}^{N} w(x_k^i)\delta(x - x_k^i) \end{aligned} \tag{1.77}$$

Eq.(1.77) indicates that the prior distribution $p(x_k|y_k)$ of any arbitrary shape can be represented using a batch of random particles associated with weights. At the prediction step of every time step, the particles of the prior distribution are propagated through the nonlinear model. The predicted distribution $p(x_{k+1}|y_k)$ is hence shaped by the predicted particles $x_{k+1}^{i,f}$. At the update step of every time step, the PF assimilates the observations by sequentially updating the weights of the particles based on the observations:

$$\tilde{w}_{k+1}^i = \tilde{w}_k^i \frac{p(x_{k+1}^i|x_k)}{q(x_{k+1}^i|x_k)} p(y_{k+1}|x_{k+1}^i) \tag{1.78}$$

In Eq.(1.78), if the importance density is selected to be the transition function, i.e. $p(x_{k+1}|x_k) = q(x_{k+1}|x_k)$, then we have $\tilde{w}_{k+1}^i = \tilde{w}_k^i p(y_{k+1}|x_{k+1}^i)$. This is called a bootstrap particle filter.

The weights are then normalized by:

$$w_{k+1}^i = \frac{\tilde{w}_{k+1}^i}{\sum_{i=1}^{N} \tilde{w}_{k+1}^i} \tag{1.79}$$

The statistic moments can be calculated using w_k^i and $x_{k+1}^{i,f}$ and this completes the sequential importance sampling (SIS) particle filtering algorithm. What the SIS algorithm essentially does is to propagate the particles through the model and sequentially update the corresponding weights. The SIS suffers from the degeneracy problem, which causes the weights of most particles to reduce to zero except for a few. In the end, very few effective particles [8] contribute to the estimation, hence the addition of the resampling step.

The resampling step aims to increase the diversity of the particles, reducing the degeneracy speed. Several resampling techniques have been proposed in literature [39][30][52]. They all attempt to generate a new batch of samples made up with duplicates of the predicted particles $x_{k+1}^{i,f}$ which potentially would contribute to a more accurate estimate. Whether or not the resampling step is carried out depends on the effective sample size, which is a measure of the concentration of samples in the region of interest. It is given by:

$$\hat{N}_{eff} \approx \frac{1}{\sum_{i=1}^{N}(w_k^i)^2} \tag{1.80}$$

If $\hat{N}_{eff}$ falls below a threshold, then the samples have been spreaded too far and the resampling step is required. The basic steps of an inverse transform method for the resampling are described as follows:

(1) Generate a discrete cumulative distribution function(CDF) based on the weights of the particles.

For $j = 1 \quad to \quad N$

$$F_j = \sum_{i=1}^{j} w_k^i \tag{1.81}$$

With N particles, F has N fragments.

(2)Generate N random numbers from a uniform distribution $[0 \quad 1]$, denoted as matrix U.

(3) For each random number, find the index j, such that $\quad F_{j-1} < U(i) < F_j$. Select $x_{k|k}^j$ to be posterior particle.

With the resampling step, the sequential importance resampling (SIR) algorithm is completed. The bootstrap particle filter is implemented as follows:

1. Initialize the filter:

$$x_0^i = q(x_0), \quad i = 1, \ldots, N$$
$$w_0^i = \frac{1}{N}$$

2. Propagate the particles:

$$x^i_{k+1|k} = f(x^i_{k|k}) + w_i \quad i = 1, \ldots, N$$

3. Calculate the normalized weights:

$$\tilde{w}^i_{k+1} = \tilde{w}^i_k p(y_{k+1}|x^i_{k+1})$$

4. Normalize the weights:

$$w^i_{k+1} = \frac{\tilde{w}^i_{k+1}}{\sum_{i=1}^N \tilde{w}^i_{k+1}}$$

5. Calculate N_{eff} as in Eq.(1.80)

6. If N_{eff} is lower than the threshold, perform the resampling as is described previously.

7. Calculate the updated mean and covariance:

$$\mu_{k+1|k+1} = \frac{1}{N}\sum_{i=1}^N x^i_{k+1|k+1}$$
$$\Sigma_{k+1|k+1} = \frac{1}{N}\sum_{i=1}^N (x^i_{k+1|k+1} - \mu_{k+1|k+1})(x^i_{k+1|k+1} - \mu_{k+1|k+1})^T$$

The particle smoother can be formulated using a forward-filtering-backward-smoothing framework [18] or a two-filter framework [53]. The former is formulated the same as the previous nonlinear RTS smoothers and the latter is formulated with two filters; one running forward and one backward. Both of them attempt to obtain the SMC approximation for $f(x_k|y_{1:k})$ given by:

$$p(x_k|y_{1:T}) = \sum_{i=1}^N w^i_{k|T}\delta(x - x^i_k) \tag{1.82}$$

Comparing Eq.(1.77) to Eq.(1.82), we can see that the smoothing reshapes the posterior distribution $p(x_k|y_{1:k})$ by updating the weights of the particles using the future observations beyond k.

The RTS particle smoother updates the backward weights by:

$$w^i_{k|T} = \sum_{i=1}^{N} w^i_k [\sum_{j=1}^{N} w^j_{k+1|T} \frac{p(x^j_{k+1}|x^i_k)}{\sum_{l=1}^{N} w^l_k p(x^j_{k+1}|x^l_k)}] \tag{1.83}$$

With simple manipulation of Eq.(1.83), we can find out that Eq.(1.83) costs $O(N^2)$ operations to evaluate. The details of the derivation of Eq.(1.83) are given in chapter 6.

The two-filter smoother updates the backward weights by:

$$w^i_{k|T} = \tilde{w}^i_k \sum_{j=1}^{N} w^j_{k-1} \frac{p(\tilde{x}^i_k|x^j_{k-1})}{\tilde{p}(\tilde{x}^j_k)} \tag{1.84}$$

where $\tilde{w}^i_k$, $\tilde{x}^i_k$ are the weights and particles newly generated in the backward filtering. $\tilde{p}(x_k)$ is an auxiliary probability density which enables us to form the analytic expression of Bayesian inference using the SMC approach. It can be an arbitrary distribution as long as it does not cause any zero divisions. Note that Eq.(1.84) also takes $O(N^2)$ operations to evaluate.

1.3 Contributions

The contribution of the thesis can be summarized in the following points:

- First, the thesis develops a novel nonlinear filter for non-Gaussian systems called the Gaussian mixture model based ensemble Kalman filter. Although this filter is formulated based on the existing work in the literature, modifications are made on the original work. Besides in-depth analysis is performed on this filter to illustrate its advantages over the existing solutions, such as the EnKF and the PF.

- Second, the thesis develops a novel constrained state estimation approach based on the Kullback-Leibler divergence. The proposed method is integrated with the EKF and the EnKF to form novel constrained EKF and constrained EnKF. The proposed constrained filters are tested through case studies and the results show their advantages over the existing constrained method, the recursive nonlinear dynamic data assimilation.

- Third, the thesis modifies the proposed Gaussian mixture model based ensemble Kalman filter into its constrained version using the Kullback-Leibler divergence. Prior to our work, almost no existing work has been found for constrained non-Gaussian filters. In order to develop this constrained filter, our work essentially solves an optimization problem to minimize the Kullback-Leibler divergence of two Gaussian mixture distributions.

- Fourth, the thesis further develops a novel Gaussian mixture model based ensemble Kalman smoother. This proposed smoother solves the smoothing problem for nonlinear non-Gaussian systems. Again very few work has been done for non-Gaussian smoothers. This smoother is developed under a RTS smoother framework and we also borrow the idea of the particle smoother to update the particle weights to further improve the proposed smoother.

1.4 Outline

The chapters in the dissertation are organized as follows:

Chapter 2 presents the proposed Gaussian mixture model based ensemble Kalman filter and its application on a MMA polymerization model.

Chapter 3 presents the proposed constrained state estimation approach using Kullback-Leibler divergence and its integration with the nonlinear Gaussian filters, EKF and EnKF. The constrained EKF and EnKF along with their applications on two CSTR processes are introduced in detail.

Chapter 4 presents the proposed constrained state estimation approach using Kullback-Leibler divergence for non-Gaussian filter, specifically the Gaussian mixture model based ensemble Kalman filter in chapter 2. The proposed filter is applied on the data collected from a real algae cultivation process.

Chapter 5 presents the proposed Gaussian mixture model based ensemble Kalman smoother. The proposed smoother is also applied on the same algae cultivation process for its performance test.

Chapter 2

Gaussian mixture model-based ensemble Kalman filter

2.1 Introduction

Over the years, the most popular estimator used in nonlinear chemical processes in general and specifically for polymerization reactors, too, is the extended Kalman filter (EKF) (e.g., [104][73][44][56][65][66][92][29]). However, this estimator involves linearization of the original model at each step, and can be inaccurate for highly nonlinear systems. Our focus in this work is on particle-based estimators, which are derivative free estimators using different sampling methods to generate an ensemble of particles to represent the distributions of the dynamic states of the system.

The most commonly used estimators based on the use of an ensemble of particles are the EnKF, the UKF and the PF. While the EnKF and the UKF provide only the mean and variance of the posterior distribution of the states since they use a Gaussian assumption for the distributions, the PF, which works on Bayesian principles, can provide estimates for the full distribution of the states even in situations where the distribution is not Gaussian (which occurs in nonlinear systems) by using a set of particles associated with different weights. In practice, the application of the PF to chemical processes is very recent. Chen et al. [14] compared the performance of the auxiliary particle filter with an EKF for a batch polymethyl methacrylate process to show that it outperformed the EKF in terms of the root mean squared error for state and parameter estimation. Shenoy et al. [84] compared the UKF, EKF and PF in a case study on a polyethylene reactor simulation to demonstrate that the PF provided more accurate estimation results, but was less robust to plant-model mismatch. Shao et al. [83] compared the performance of the PF, EKF, UKF and moving horizon estimation for constrained state estimation and showed that the constrained PF provided more accurate estimation results compared to other methods.

An important issue with the PF relates to its performance for high dimensional systems. The EnKF, on the other hand, has the advantage of being scalable to high-dimensional systems without a prohibitive increase in the size of the ensemble required; however, as is stated earlier, the algorithm is based on the assumption that both the prior and posterior distribution of the states can be approximated by the Gaussian distribution, and it may be unreliable when this assumption is not valid.

In this chapter, we consider the application of the nonlinear filters on non-Gaussian systems. Specifically we focus on the polymerization process. The polymerization processes offer unique challenges for process modeling, monitoring and control. The production of polymers of different grades means that the process conditions are changed relatively frequent. Product quality specifications (usually expressed in terms of constraints on the properties of the molecular weight distribution) and dynamic operation lead to the need for on-line monitoring and control, which need accurate process models and real-time estimation of states and parameters of the system.

Polymerization processes can be of high dimension when they are described using population balance models [15][51], and a multimodal distribution of properties such as the particle size and molecular weight may be desirable [81][20][26]. This, especially in the presence of model-plant mismatch, creates challenges for both the EnKF and the PF. Also, the nonlinearity of the systems may lead to multimodality in the state distributions. In this chapter, we propose a Gaussian mixture model based ensemble Kalman filter using the polymerization process as an example. We present results on the application of this filter to a polymethyl methacrylate (PMMA) process and compare its performance to that of the EnKF and the PF.

2.2 Review of the existing mixture ensemble filters

An appropriate model for the prior distribution plays an importance role in achieving an accurate estimate. The idea of using a Gaussian mixture model(GMM) for the prior distribution has been considered long ago. For example, in [1], a Gaussian sum filter was proposed to extend the standard Kalman update to the Gaussian mixture distributions. The crucial problem is how to update the parameters of the Gaussian mixture model when assimilating the observations. In [1], the mean and covariance of each Gaussian mode in the mixture can be updated directly using the Kalman gain for linear systems. However, the problem is trickier for nonlinear systems. In this section, we solve this problem by combining the idea of GMM with Monte Carlo representation of the state PDF. Such filters are referred to as mixture ensemble filter. In the following, we first review the some of the most prominent developments on mixture ensemble filters for nonlinear systems in recent years.

Bengtsson et al.(2003)

The algorithm of the mixture ensemble filter proposed by Bengtsson et al.[7] is as follows.

First, a GMM is derived from the predicted ensemble $\{x_k^{f,i}\}_{i=1,\ldots,N}$. M random members $\{x_k^{f,l}\}_{l=1,\ldots,M}$ are selected from this ensemble as the centroids of each Gaussian mode. Then for each centroid, select m members in the ensemble which have the closest distance to the centroid point based on the Euclidean norm. The predicted mean and covariance can be calculated through the M ensemble members in each cluster. Next, update every member in each cluster using as follows:

$$x_k^{a,i} = x_k^{f,i} + K_j(y_k - Hx_k^{f,i}) \quad i = 1, \ldots, N; \;\; j = 1, \ldots, M \tag{2.1}$$

where K_j is the Kalman gain of each cluster j. H is the measurement operator.

One key parameter in mixture ensemble filter is the membership probability w_{ij}, which describes the probability of each member belonging to each cluster. In this work, for each cluster, $w_{ij} = \frac{1}{M}$. The updated mean $\mu_k^{a,j}$ and covariance $P_k^{a,j}$ are calculated through the samples of each cluster. Finally, the posterior distribution is given by:

$$p(x_k^a) = \sum_{j=1}^{M} \pi_k^{a,j} N(x; \mu_k^{a,j}, P_k^{a,j}) \tag{2.2}$$

The drawback of the method is that the clusters do not necessarily cover the whole ensemble. There might be members left out in two clusters, which causes the assignment of w_{ij} inaccurate. The fundamental cause is the crude clustering approach to build the GMM based upon the nearest neighbors. In their work, the authors also discuss the difficulties encountered in high-dimensional space and propose an approach to decompose the high-dimensional system into lower-dimensional system which can be handled by the mixture ensemble filter.

Smith(2007)

In [90], Smith proposed to describe the underlying structure of the predicted distribution by using the expectation maximization (EM) algorithm. The EM algorithm provides more accurate clustering results for the GMM than the heuristic clustering algorithm in Bengtsson's work. Before clustering, mixture complexity is first accessed using Akaike's information criterion(AIC). The predicted distribution from the EM clustering is given as:

$$p(x_k^f) = \sum_{j=1}^{M} \pi_k^{f,j} N(x; \mu_k^{f,j}, P_k^{f,j}) \tag{2.3}$$

where M is the number of the modes.

After obtaining the predicted GMM, Smith updates the parameters of it using the same strategy as the previous approach - by updating individual particle $x_k^{f,i}$ and its membership weight w_{ij}. However, the update is different from the described in Eq.(2.1). In Eq.(2.1) each particle is updated once using its corresponding Kalman gain calculated from the neighborhood samples, while in Smith's work each particle is updated M times by each of the cluster. This is because the EM clustering does not give the membership of each particle associated with each cluster. In order for the EnKF update to be applied, the whole ensemble is assumed to belong to one mode with a probability w_{ij} assigned to each particle. Each particle is updated under this mode using:

$$x_{k,j}^{a,i} = x_k^{f,i} + K[j](y_k - Hx_k^{f,i}) \quad j = 1, \ldots, M \tag{2.4}$$

After acquiring the M updated ensembles, Smith chooses to return to one ensemble of size N which follow a Gaussian posterior distribution, i.e. he approximates the posterior distribution with a Gaussian distribution. First the predicted ensemble associated with each mode $x_{k,j}^{f,i}$ is projected to an ensemble of a standard normal distribution by using:

$$x_{k,j}^{sn,i} = [S_{k,j}^a]^{-1}(x_{k,j}^{f,i} - \mu_{k,j}^a) \tag{2.5}$$

where $\mu_{k,j}^a$ and $P_{k,j}^a$ are the updated mean and covariance of each mode,$S_{k,j}^a = \sqrt{P_{k,j}^a}$. Then M ensemble is combined into one using:

$$x_k^{sn,i} = \sum_{j=1}^{M} x_{k,j}^{sn,i} \tag{2.6}$$

Then the normalized particles are projected to become the samples taken from a Gaussian distribution with mean μ_k^a and covariance P_k^a.

$$x_k^{a,i} = \sum_{j=1}^{M} \pi_{k,j}^a (\mu_{k,j}^a + S_{k,j}^a x_k^{sn,i}) \tag{2.7}$$

The multimodal ensemble Kalman filter proposed in this chapter is based on Smith's work. We borrow his idea of giving one mode the ownership of all particles. However, the posterior distribution in our work remains to be a Gaussian mixture distribution, i.e. the mixture feature is retained for the posterior distribution.

Dovera et al.(2010)

In [19], Dovera et al. proposed a mixture ensemble Kalman filter similar to Smith's work. In their work, the EM clustering is also performed on the predicted ensemble to obtain the underlying mixture structure of the predicted distribution. However, they use a different approach in the update step. Again the membership of each particle is crucial, Without this information, the standard EnKF update cannot be performed. Instead of assuming all particles to be owned by one mode, they solve the membership problem by directly assigning each particle to a specific mode based on the posterior mixture weights $\pi_k^{a,j}$.

The update on each of the predicted particle is performed as follows.

For each of the predicted particle $x_k^{f,i}$:

- Set r as the known component of $x_k^{f,i}$.
- Determine the index of the cluster it belongs to after the update by generating a random number from $l \in \{1, \dots, M\}$ based on the posterior mixture weights $\{\pi_k^{a,1}, \dots, \pi_k^{a,M}\}$.
- Compute an auxiliary state vector $x_k^{f,i'}$ using:

$$x_k^{f,i'} = \mu_k^{f,l} + S^l (S^r)^{-1} (x_k^{f,i} - \mu_k^{f,r}) \tag{2.8}$$

where $S^l = \sqrt{P_k^{f,l}}$ and $S^r = \sqrt{P_k^{f,r}}$. What Eq.(2.8) essentially does is to move one particle from one mode with mean and covariance $\mu_k^{f,l}, P_k^{f,l}$ to another mode with mean and covariance $\mu_k^{f,r}, P_k^{f,r}$.

- Update the particles under the cluster it belongs.

$$x_k^{a,i} = x_k^{f,i'} + K_j(y_k - Hx_k^{f,i'}) \tag{2.9}$$

where

$$K_j = P_k^{f,j} H^T (HP_k^{f,j} H^T + R)^{-1} \tag{2.10}$$

2.3 Multimodal ensemble Kalman filter(GMM-EnKF)

As is introduced in the previous section, the generic mixture ensemble consists of two steps. Firstly, a clustering analysis is performed to achieve a multimodal distribution for the state space. Secondly, the parameters of this multimodal distribution are updated in some fashion

2.3.1 Expectation maximization for clustering of Gaussian mixture model

The GMM-EnKF proposed in this work uses the EM algorithm for clustering. The PDF of a d-dimensional random vector x following a Gaussian mixture distribution is given by:

$$p_x(x) = \sum_{j=1}^{N_c} \pi_j \times N(x; \mu_j, P_j) \tag{2.11}$$

where π_j, μ_j and P_j represent the mixing weights, mean and covariance of each mode j. N_c is the number of the modes. $N(.)$ is a Gaussian with the PDF of:

$$N(x; \mu_j, P_j) = \frac{1}{(2\pi)^{d/2}|P_j|^{1/2}} e^{-\frac{1}{2}(x-\mu_j)^T P_j^{-1}(x-\mu_j)}$$

π_j are subject to constraints that:

$$0 \leq \pi_j \leq 1 \quad \text{and} \quad \sum_{j=1}^{Nc} \pi_j = 1$$

To fit a given set of data $\{x_i\}_{i=1,\cdots,N}$ to a Gaussian mixture model, the EM algorithm is used to estimated the parameters of the GMM, $\theta = \{\pi_1, \cdots, \pi_{N_c}, \mu_1, \cdots, \mu_{N_c}, P_1, \cdots, P_{N_c}\}$[17]. EM is a variant of the maximum likelihood estimation when there exists hidden variables or missing data. In the case of GMM clustering, the mode identity of each data point is considered as the missing or hidden variable. Let $\{(c_i)_j\}$ be a binary indicator vector representing the identity of the component j that generates x_i. Its value is given by:

$$(c_i)_j = \begin{cases} 1 & \text{if data point is generated by component j} \\ 0 & \text{otherwise} \end{cases}$$

In the EM algorithm, an E-step is performed first to compute the Q function, the expectation of the log likelihood of the complete data set, by first computing the probability of each data x_i belonging to each component j even the current parameters θ^k estimated from the previous iteration. The Q function is given by:

$$Q(\theta|\theta^k) = E[L(p(z|\theta))|x, \theta^k] \tag{2.12}$$

where L takes the log likelihood of the subjective function. x is the observed data; z is the complete data set consisting of both observed z and missing data $(c_i)_j$; θ^k is the estimate of the last iteration. The Q function can further be written as:

$$Q(\theta|\theta^k) = \sum_{i=1}^{N} \sum_{j=1}^{N_c} p[(c_i)_j|\{x\}, \theta^k](log \pi_j N(x_i; \mu_j, P_j)) \tag{2.13}$$

$p[(c_i)_j|\{x\}, \theta^k]$ in Eq.(2.13) is the key step in calculating the Q function. It is called the membership weight which calculates the probability of data x_i belonging to the mode j

given parameters θ^k. The E step of the EM algorithm calculates this membership weight as follows.

$$w_{ij} = p[(c_i)_j|\{x\}, \theta^k] = \frac{\pi_j^k N(x_i; \mu_j^k, P_j^k)}{\sum_{m=1}^{N_c} \pi_m^k N(x_i; \mu_m^k, P_m^k)} \tag{2.14}$$

In the M step, the Q function in Eq.(2.13) is maximized with respect to θ, i.e. π_j, μ_j and P_j, to estimate the θ^{k+1}. The estimated θ^{k+1} is given as follows.

$$\pi_j^{k+1} = \frac{N^k}{N} \tag{2.15}$$

$$\mu_j^{k+1} = \frac{1}{N^k} \sum_{i=1}^{N} w_{ij} x_i \tag{2.16}$$

$$P_j^{k+1} = \frac{1}{N^k} \sum_{i=1}^{N} w_{ij} (x_i - \mu_j^{k+1})(x_i - \mu_j^{k+1})^T \tag{2.17}$$

where $N^k = \sum_{i=1}^{N} w_{ij}$.

The E-step and the M-step are performed iteratively until the estimates converge. During this process, the problem of singularity may arise when one of the components collapses onto one data point. This usually happens due to over-fitting in the maximum likelihood estimation(MLE). To avoid this problem, we use a modified update for the covariance given by:

$$P_j^{k+1} = \frac{\sum_{i=1}^{N} w_{ij} (x_i - \mu_j^{k+1})(x_i - \mu_j^{k+1})^T + \lambda I_d}{N^k + 1} \tag{2.18}$$

where I_d is an n-dimensional unit matrix and λ is a regularization constant determined by some validation data[95]. An alternate(ad hoc) method to deal with the problem of singularity is to detect when the singularity occurs and reset the means of all components randomly and the covariance to some larger value.

The pseudo-code for the EM algorithm is provided below.

Algorithm 2.1: Expectation maximization algorithm. Inputs are data set $\{x_i\}$. component number N_c and initial values $\{\theta^0\}$ of $\{\pi_j\}_{j=1,\cdots,N_c}$,$\{\mu_j\}_{j=1,\cdots,N_c}$, $\{P_j\}_{j=1,\cdots,N_c}$, initially $\theta^k = \theta^0$

```
$\{\theta^{k+1}\} = EM[\{x\}, N_c, \{\theta^k\}]$
// E step
while $\epsilon \leq 1e-6$
   for $i = 1 : N$
      for $j = 1 : N_c$
         $p[(c_i)_j|x_i, \theta^k] = \frac{\pi_j^k N(x_i;\mu_j^k,P_j^k)}{\sum_{m=1}^{N_c} \pi_m^k N(x_i;\mu_m^k,P_m^k)}$
      end for
   end for
// M step
   for $j = 1 : N_c$
      $\pi_j^{k+1} = \sum_{i=1}^{N} P[(c_i)_j, \theta^k]/N$
      $\mu_j^{k+1} = \sum_{i=1}^{N} P[(c_i)_j, \theta^k]x_i/\sum_{i=1}^{N} P[(c_i)_j, \theta^k]$
      $P_j^{k+1} = (\sum_{i=1}^{N} P[(c_i)_j, \theta^k](x_i - \mu_j^{k+1})(x_i - \mu_j^{k+1})^T + \lambda I_d)/(\sum_{i=1}^{N} P[(c_i)_j, \theta^k] + 1)$
   end for
   $\epsilon = \mu^{k+1} - \mu^k$
end while
return $\theta^{k+1}$
```

2.3.2 GMM-EnKF filtering algorithm

In this section, a GMM-based EnKF (GMM-EnKF) filter is proposed to obtain estimates of the full state distribution. As with the EnKF and PF, the GMM-EnKF also uses a set of Monte Carlo samples to present the posterior PDF of the states which is approximated with the GMM at every time step.

At each time step k, the GMM-EnKF has two steps-forecast and update. The forecast step is identical to the EnKF. An ensemble of size N, $\{x_k^i\}_{i=1,\cdots,N}$, is drawn from the prior distribution of the states and forwarded through the model to obtain a predicted ensemble for the next time step. Then, the EM algorithm is performed on the predicted ensemble to obtain the estimates of the GMM with N_c components. Next, the Kalman update is performed based on each component in the GMM to get an ensemble of size $N \times N_c$. Finally,these ensemble members are combined based on their weights and reduced to a size of N. The details of the algorithmic sequence are as follows:

Forecast:

1. The first portion of the forecast step is to determine the number of components N_c in the multimodal distribution. N_c can be determined using the Bayesian or other information criteria(BIC) [41], or using prior knowledge. For example, in reservoir models, petrophysical properties such as porosity or permeability are typically related to geological units (facies) and variables inside the facies are characterized by underlying multimodal distributions which are known beforehand [23]. In our work, this information can be considered as prior knowledge if we know the distribution of the process noise.

2. With the knowledge of the process model and the number of components N_c, the prior ensemble $\{x_k^i\}_{i=1,\cdots,N}$ is propagated through the model to get the values of the predicted ensemble $\{x_k^{f,i}\}_{i=1,\cdots,N}$. These particles can be considered as the realizations of the predicted state space x^f. Assume the predicted state distribution $p(x_k^f)$ is a GMM, it is given by:

$$p(x_k^f) = \sum_{j=1}^{N_c} \pi_{k,j}^f p_j(x_k^f) = \sum_{j=1}^{N_c} \pi_{k,j}^f N(x_k^f; \mu_{k,j}^f, P_{k,j}^f) \tag{2.19}$$

The EM algorithm is applied on $\{x_k^{f,i}\}_{i=1,\cdots,N}$ to give us the parameters of the predicted distribution, $\pi_{k,j}^f, \mu_{k,j}^f \text{and} P_{k,j}^f$, of each component j.

Update:

3. For each component j of the distribution, the Kalman gain matrix for each Gaussian component is computed by utilizing the membership probability matrix W.

$$P[j]^f H^T = \sum_{i=1}^{N} w_{i,j}(x_k^{f,i} - \mu_{k,j})(Hx_k^{f,i} - H\mu_{k,j})^T / n_j \tag{2.20}$$

$$HP[j]^f H^T = \sum_{i=1}^{N} w_{i,j}(Hx_k^{f,i} - H\mu_{k,j})(Hx_k^{f,i} - H\mu_{k,j})^T / n_j \tag{2.21}$$

$$K[j] = P[j]^f H^T (HP[j]^f H^T + R)^{-1} \tag{2.22}$$

where $w_{i,j} = \frac{\tau_{k,j}^f N(x_k^{f,i}; \mu_{k,j}^f, P_{k,j}^f)}{\sum_{m=1}^{N_c} \tau_{k,m}^f N(x_k^{f,i}; \mu_{k,m}^f, P_{k,m}^f)}$, $n_j = \sum_{i=1}^{N} w_{i,j}$ and H is the linearized measurement operator.

4. In the update step, assume that one Gaussian component j claims the ownership of all the ensemble members, the Kalman update can be performed for each particle i under each mode j. This gives us an ensemble size of $N \times N_c$.

$$x_{k,j}^{a,i} = x_k^{f,i} + K[j](d - Hx_k^{f,i} - v_i) \tag{2.23}$$

where v_i is the measurement noise.

5. The $N \times N_c$ ensemble members can be combined to form N members by using the probability matrix W. This gives us the final posterior ensemble $\{x_k^{a,i}\}_{i=1,\cdots,N}$.

$$x_k^{a,i} = \sum_{j=1}^{N_c} w_{i,j} x_{k,j}^{a,i} \tag{2.24}$$

6. The mean and covariance of the posterior distribution can be computed as

$$\mu_{k,j}^a = \sum_{j=1}^{N} w_{i,j} x_{k,j}^{a,i}/n_j \tag{2.25}$$

$$P_{k,j}^a = \sum_{j=1}^{N} w_{i,j}(x_{k,j}^{a,i} - \mu_{k,j}^a)(x_{k,j}^{a,i} - \mu_{k,j}^a)^T/n_j \tag{2.26}$$

7. The posterior weight of each component of the distribution can be computed based on the observed data d, which contains the measurements y.

$$\pi_{k,j}^a = p(\mu_{k,j}^a, P_{k,j}^a, R|d) = \frac{p(d|\mu_{k,j}^a, P_{k,j}^a, R)n_j}{\sum_{m=1}^{N_c} p(d|\mu_{k,j}^a, P_{k,j}^a, R)n_j} \tag{2.27}$$

$$p(d|\mu_{k,j}^a, P_{k,j}^a, R) = \frac{exp[-\frac{1}{2}(d - H\mu_{k,j}^a)^T(HP_{k,j}^a H^T + R)^{-1}(d - H\mu_{k,j}^a)]}{\sqrt{(2\pi)^m |HP_{k,j}^a H^T + R|}} \tag{2.28}$$

8. With the estimates of the parameters of the GMM $\mu_{k,j}^a, P_{k,j}^a$ and $\pi_{k,j}^a$, we have a full distribution of the posterior state distribution. The point estimate at time k given by the

posterior PDF is given by:

$$x_k^a = \sum_{j=1}^{N_c} \pi_{k,j}^a \mu_{k,j}^a \tag{2.29}$$

The pseudo-code for the GMM-EnKF algorithm is provided below:

Algorithm 2.2: GMM-EnKF algorithm. Inputs include the initial distribution of x, the total number of the particles N, the components N_c, and the time steps T. Inputs and measurements of the system at each time step are u_k and d_k.

```
[$\{x_k^{a,i}\}, \{\pi_{k,j}^a, \mu_{k,j}^a, P_{k,j}^a\}$]=GMM-EnKF[$\{x_k^i\}, \{u_k\}, \{d_k\}$]

for $k = 1 : T$
  for $i = 1 : N$
    Draw $\{x_k^i\}_{i=1,\cdots,N} \sim f(x_k)$
    Calculate $x_k^{f,i} = f(x_k^i, u_k, w_k^i)$
    Calculate $y_k^i = Hx_k^i + v_k^i$.
  end for
  Apply the EM algorithm on $\{x_k^{f,i}\}_{i=1,\cdots,N}$ using the algorithm 1:
  $\{\pi_{k,j}^f, \mu_{k,j}^f, P_{k,j}^f\}_{j=1,\cdots,N_c} = EM[\{x_k^{f,i}\}, N_c, \theta^k]$
  for $j = 1 : N_c$
    Calculate the Kalman gain of each component $K[j]$ using Eq.(2.22).
    for $i = 1 : N$
      Calculate the updated particle under each component $x_{k,j}^{a,i}$ using Eq.(2.23).
    end for
    Combine $\{x_{k,j}^{a,i}\}_{i=1,\cdots,N}$ to obtain the posterior particles $\{x_k^{a,i}\}_{i=1,\cdots,N}$ using Eq.(2.24).
    Calculate the parameters of the posterior distribution $\pi_{k,j}^a, \mu_{k,j}^a, P_{k,j}^a$ using Eq.(2.25)
    to Eq.(2.27).
  end for
  Calculate the posterior point estimate $x_k^a$ using Eq.(2.29).
end for.
```

While the PF and the GMM-EnKF both can, in principle, account for multimodality, the use of the Gaussian mixture model provides the GMM-EnKF with greater flexibility in capturing a wide variety of distributions under varying levels of model-plant mismatch, as will be shown in the following results.

2.4 Simulations and discussion

2.4.1 Mathematical model of the MMA polymerization process

Simulations of a free-radical MMA polymerization process are used to demonstrate the performance of the estimation method proposed in this paper. The process is assumed to take place in a continuous stirred tank reactor (CSTR) and uses AIBN as the initiator and toluene as the solvent. The mathematical model of this process is described below in Eq.(2.30) to Eq.(2.36), and further details can be found in [87][85]. The six states to be estimated include the monomer concentration C_M, the initiator concentration C_I, the reactor temperature T, the moments of the polymer distribution D_0 and D_1, and the jacket temperature T_j. Only the temperatures are measured. The number average molecular weight (NAMW), which is the primary quality variable for the process, is defined as the ratio D_1/D_0.

$$\frac{dC_M}{dt} = -(k_p + k_{fm})C_M P_0 + \frac{F(C_{Min} - C_M)}{V} \tag{2.30}$$

$$\frac{dC_I}{dt} = -k_I C_I + \frac{F_I C_{Iin} - F C_I}{V} \tag{2.31}$$

$$\frac{dT}{dt} = \frac{-\Delta H k_p C_M P_0}{\rho C_\rho} - \frac{UA}{\rho C_\rho V}(T - T_j) + \frac{F(T_{in} - T)}{V} \tag{2.32}$$

$$\frac{dD_0}{dt} = (0.5k_{tc} + k_{td})P_0^2 + k_{fm} C_M P_0 - \frac{F D_0}{V} \tag{2.33}$$

$$\frac{dD_1}{dt} = M_m(k_p + k_{fm})C_M P_0 - \frac{F D_1}{V} \tag{2.34}$$

$$\frac{dT_j}{dt} = \frac{F_{cw}(T_{w0} - T_j)}{V_0} + \frac{UA}{\rho_w C_{pw} V_0}(T - T_j) \tag{2.35}$$

$$P_0 = \sqrt{\frac{2f^* + C_I k_I}{k_{td} + k_{tc}}} \tag{2.36}$$

In all the simulations whose results are described in the following sections, the number of particles used for each estimator, N, is 100. The number of components, N_c, is set to 2. The parameters of the bi-modal noise in all simulations are $\mu = [0.1, 0.8]$, $P = diag(0.1, 0..1)$ for states C_M,C_I and D_0; $\mu = [8, 64]$, $P = diag(8, 8)$ for state D_1; and $\mu = [0.6, 4.8]$, $P = diag(0.6, 0.6)$ for states T and T_j.

Table 2.1: Operational parameters for the MMA polymerization reactor

$F = 1.0m^3/h$	$M_m = 100.12kg/kgmol$
$F_I = 0.0032m^3/h$	$F^* = 0.58$
$F_{cw} = 0.1588m^3/h$	$R = 8.314KJ/kgmol \cdot K$
$C_{Min} = 6.4678kgmol/m^3$	$-\Delta H = 57800KJ/kgmol$
$C_{Iin} = 8.0kgmol/m^3$	$E_p = 1.8283 \times 10^4 KJ/kgmol$
$T_{in} = 350K$	$E_I = 1.2877 \times 10^5 KJ/kgmol$
$T_{w0} = 293.2K$	$E_{fm} = 7.4478 \times 10^4 KJ/kgmol$
$U = 720KJ/h \cdot K \cdot m^2$	$E_{tc} = 2.9442 \times 10^4 KJ/kgmol$
$A = 2.0m^2$	$E_{td} = 2.9442 \times 10^4 KJ/kgmol$
$V = 0.1m^3$	$A_p = 1.77 \times 10^9 m^3/kgmol \cdot h$
$V_0 = 0.02m^3$	$A_I = 3.792 \times 10^{18} 1/h$
$\rho = 866kg/m^3$	$A_{fm} = 1.0067 \times 10^{15} m^3/kgmol \cdot h$
$\rho_w = 1000kg/m^3$	$A_{tc} = 3.8223 \times 10^{10} m^3/kgmol \cdot h$
$C_p = 2.0KJ/(kg \cdot K)$	$A_{td} = 3.1457 \times 10^{11} m^3/kgmol \cdot h$
$C_{pw} = 4.2KJ/(kg \cdot K)$	

The simulations we perform are introduced here: case study 1 provides a comparison of the GMM-EnKF, the PF and the EnKF for a case with bi-modal distributions and insignificant model-plant mismatch. Case study 2 provides a comparison of the three estimators where the model-plant mismatch is significant. Case study 3 compares the estimators for state estimation with uncertain parameters, but with the uncertain parameter not being estimated. Case study 4 considers the same case as case study 3, but with combined state and parameter estimation. In case study 5, we consider an alternate version of the PF and use the simulation conditions of case study 2.

2.4.2 Comparison of state estimation with the GMM-EnKF, EnKF and PF (case study 1 and case study 2)

In this section, we present the results of applying the GMM-EnKF, EnKF and PF algorithms on the PMMA process. To illustrate the performance of the estimators in cases where the states have multimodal distributions, bimodal process noise is applied to all the six states. The measurement noise is assumed to be Gaussian. The prior distribution of the state is also assumed to follow a GM distribution which contains two modes.

In *case study 1*, the true initial values of the states are:

$$x_0 = [5\ \ kgmol/m^3, 3\ \ kgmol/m^3, 320, 0.5\ \ kgmol/m^3, 0.5\ \ kg/m^3, 300K] \tag{2.37}$$

The dynamics of the simulation describe how the system relaxes to a steady state from this initial condition. For the estimators, the initial particles are drawn from the prior distribution. The tuning parameters for the prior distribution are its mean and covariance. In the first case, a prior distribution with a small amount of bi-modal process noise is tested for the three algorithms. The means of the two Gaussian modes of the prior distribution are:

$$\mu_1 = [4\ \ kgmol/m^3, 2\ \ kgmol/m^3, 310K, 0.49\ \ kgmol/m^3, 0.49\ \ kg/m^3, 295K];$$
$$\mu_2 = [6\ \ kgmol/m^3, 4\ \ kgmol/m^3, 330K, 0.51\ \ kgmol/m^3, 0.51\ \ kg/m^3, 305K]$$

The covariances of the modes of the prior distribution are:

$$P_1 = diag(4, 4, 28, 0.8, 8 \times 10^{-4}, 6);$$
$$P_2 = diag(4, 4, 28, 0.8, 8 \times 10^{-4}, 6)$$

The tuning parameters of the initial distribution indicate a state distribution with insignificant bimodality. The purpose of this simulation is to demonstrate the estimation performance of the three algorithms in the scenario where the state distribution shows insignificant multimodality.

The comparison of estimation results using the EnKF-GMM, EnKF and PF is shown in Figure 2.1, with time steps on the x-axis (each time step is 0.3 hr=18 min). Table 2.2 shows the root mean squared error (RMSE) over the 25 time steps of the simulation for the six states and the NAMW for the three algorithms. In this case, the estimation results from figure 1 and table 1 show that the three algorithms have similar performance in estimation of the six states. However, the EnKF-GMM has the best performance in the estimation of the NAMW. In addition, the converged variance of the estimates of the states, obtained from the estimated covariance matrix with the GMM-EnKF, are $[1\times10^{-4}, 1\times10^{-4}, 1.2\times10^{-4}, 1\times10^{-5}, 2\times10^{-4}, 4\times10^{-4}]$, respectively, confirming the significance of the estimates. The PF performs better than the EnKF only for some states. Increasing the number of particles for each of the algorithms to 200 (results not shown) improves the performance of the PF slightly, but the same conclusions hold.

Table 2.2: RMSE of the EnKF-GMM, EnKF and PF for the PMMA process with multimodal process noise (case study 1).

Variable	GMM-EnKF	EnKF	PF
$C_M, kgmol/m^3$	0.20	0.20	0.33
$C_I, kgmol/m^3$	0.24	0.20	0.33
T, K	4.3	4.4	3.1
$D_0, kgmol/m^3$	0.019	0.014	0.032
$D_1, kg/m^3$	11.85	11.53	10.44
T_j, K	2.3	2.2	1.4
NAMW	209	338	357

In *case study 2*, the multimodal features of the prior distribution are made more significant compared with the first case. The parameters of the prior distribution given below indicate that both modes lie far away from the true value, which also means that the initial condition mismatch is much larger. The true initial values of the states remain the same as the first case, and the process noise and measurement noise applied to the plant remain unchanged as well. The modified prior distribution is specified by:

$$\mu_1 = [1\ \ kgmol/m^3, 1\ \ kgmol/m^3, 290K, 0.49\ \ kgmol/m^3, 0.49\ \ kg/m^3, 270K];$$

$$\mu_2 = [10\ \ kgmol/m^3, 8\ \ kgmol/m^3, 350K, 0.51\ \ kgmol/m^3, 0.51\ \ kg/m^3, 330K];$$

$$P_1 = diag(0.8, 0.8, 5.6, 8\times10^{-2}, 8\times10^{-3}, 5.6);$$

$$P_2 = diag(0.8, 0.8, 5.6, 8\times10^{-2}, 8\times10^{-3}, 5.6));$$

In this case, the parameters of the prior distribution indicate that both of the modes lie

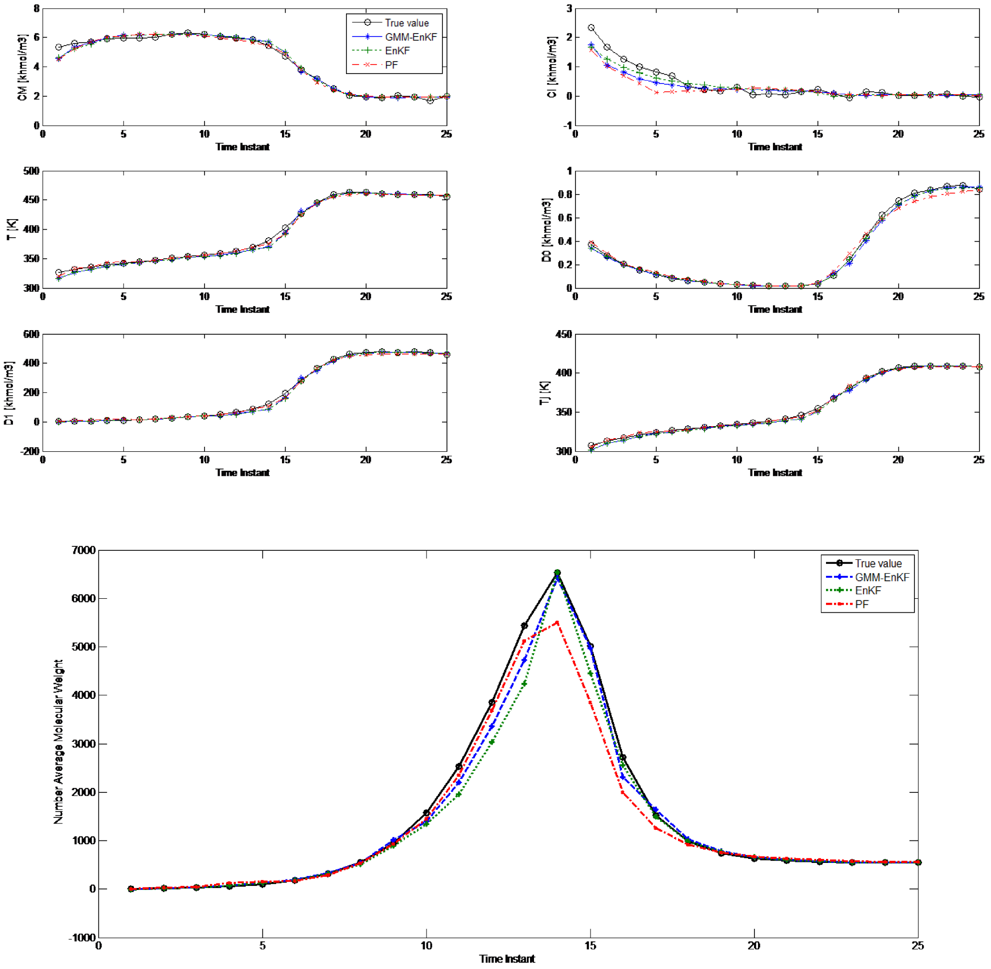


Figure 2.1: Comparison of the estimation performance of the EnKF-GMM, EnKF and PF for the PMMA process with multimodal process noise (case study 1).

near the tail of the likelihood function. The initial particles not only show significant multimodality, but also some degree of model-plant mismatch. The comparison of estimation using the EnKF-GMM, EnKF and PF is shown in figure 2.2 and the RMSE is shown in table 2.3, and it is clear that the EnKF- GMM outperforms the other two estimators. As is expected, the performance of the EnKF has worsened in this case because its Gaussian assumption on the prior and posterior distributions is violated in a significant manner. The PF does not show good performance either, and it is outperformed by the EnKF in the estimation of the NAMW. This is because the PF lacks robustness to plant-model mismatch[84],which is present in this case. Increasing the number of particles for all the estimators does not change these conclusions.

Table 2.3: RMSE of the GMM-EnKF, EnKF and PF for the PMMA process with more significant multimodal process noise (case study 2).

Variable	GMM-EnKF	EnKF	PF
$C_M, kgmol/m^3$	0.44	0.68	0.69
$C_I, kgmol/m^3$	0.37	0.14	0.17
T, K	5.8	11.8	14.4
$D_0, kgmol/m^3$	0.042	0.062	0.078
$D_1, kg/m^3$	9.73	36.13	51.38
T_j, K	5.1	8.2	9.2
NAMW	559	1400	831

Figure 2.3 shows the evolution of the multimodal posterior distribution of the one of the states (the monomer concentration) at time steps 1, 3, 4,9. Table 2.4 lists the corresponding estimation errors of the three algorithms at those time steps with respect to the true value of C_M. Figure 2.4 shows the evolution of the posterior distribution of another state (the jacket temperature) at time steps 2, 6, 9, 10 and table 2.5 shows the corresponding estimation errors of the three algorithms. These distributions are bi-modal, which clearly shows that the EnKF-GMM outperforms the other estimators in the presence of multimodal distributions.

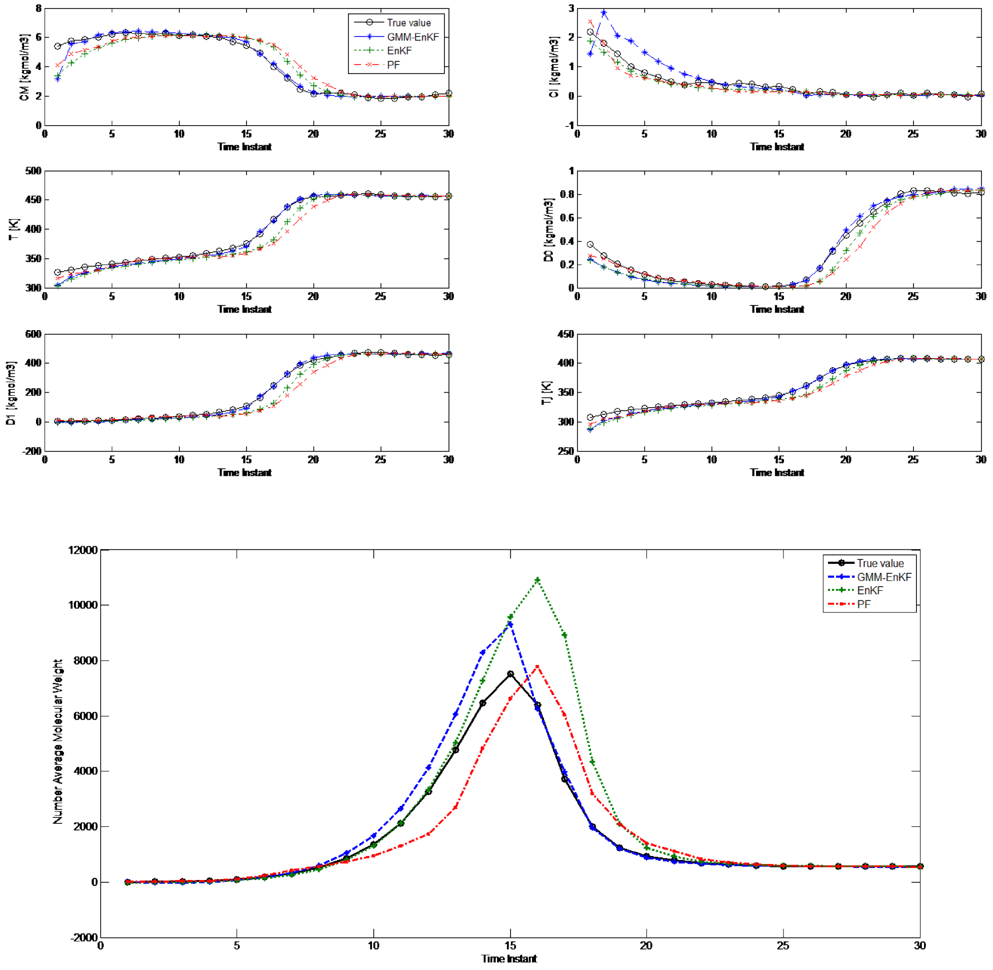


Figure 2.2: Comparison of the estimation performance of the GMM-EnKF, EnKF and PF for the PMMA process with more significant multimodal process noise (case study 2).

Table 2.4: Comparison of the estimation errors of the GMM-EnKF, EnKF and PF for C_M at time steps 1, 3, 4, 9 (in kgmol/m3).

Estimator	Time step 1	Time step 3	Time step 4	Time step 9
GMM-EnKF	0.23	0.14	0.40	0.04
EnKF	2.06	1.06	0.80	0.10
PF	3.60	2.20	1.65	0.22

Table 2.5: Comparison of the estimation errors of the GMM-EnKF, EnKF and PF for T_j at time steps 2, 6, 9, 10 (in K).

Estimator	Time step 1	Time step 3	Time step 4	Time step 9
GMM-EnKF	6.4	2.8	1.5	1.3
EnKF	6.6	3.0	1.9	1.7
PF	13.5	4.5	2.9	2.9

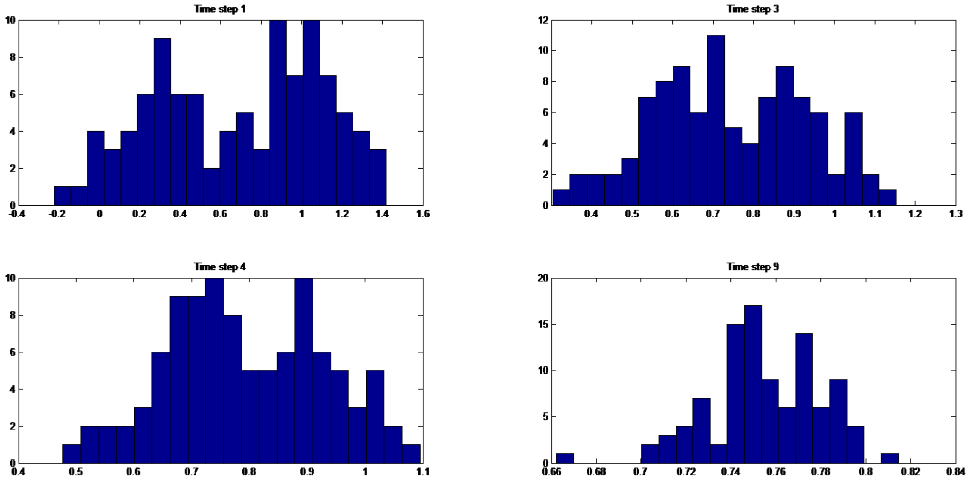


Figure 2.3: Evolution of the multimodal posterior distributions of C_M at time steps 1, 2, 4, 9.

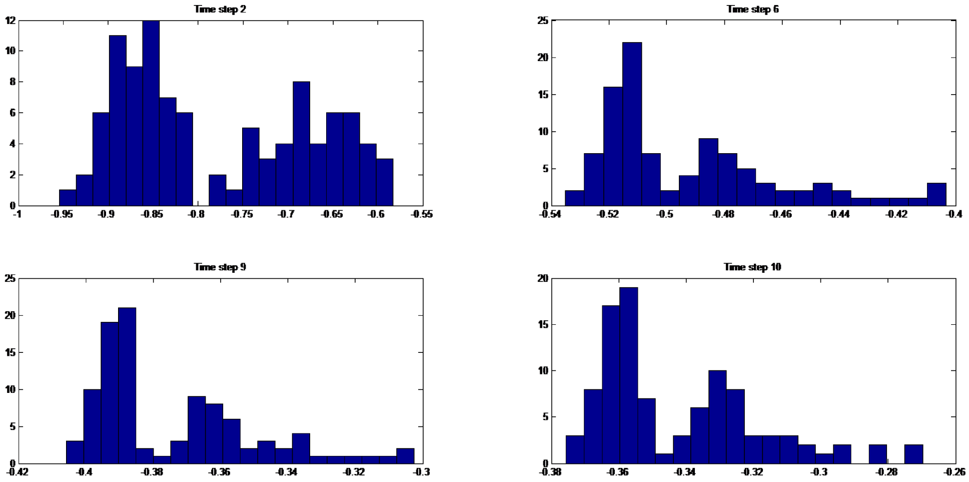


Figure 2.4: Evolution of the multimodal posterior distributions of T_j at time steps 2, 6, 9, 10.

2.4.3 Comparison of state and parameter estimation with the EnKF-GMM, EnKF and PF (case studies 3 and 4)

We consider the effects of parametric uncertainty in this section. The uncertain parameter chosen for these studies is E_p, which is the activation energy associated with the reaction rate parameter k_p. We choose E_p as the uncertain parameter because (based on dimensionless sensitivity analysis) the NAMW is highly sensitive to the values of this parameter. We consider state estimation and joint state and parameter estimation in this section.

2.4.3.1 State estimation with uncertain parameter (case study 3)

In this sub-section, while E_p is an uncertain parameter and noise is added to its value at each time step in the simulation, the parameter is not estimated. The nominal value of E_p is set to be $E_p = 1.8283 \times 10^4 kJ/kgmol$ and bi-modal Gaussian noise with means of the modes $\mu = [-100, 100]$ and covariances $P = diag(50, 50)$ is added to it. In addition, process and measurement noise with the same distributions as in the second case in the previous section are included. Figure 2.5 shows the comparison of the estimation results using the three algorithms over 40 time steps and Table 2.6 shows the corresponding RMSE. In this case, the GMM-EnKF shows a small improvement in state estimation performance over the other estimators, especially in the estimation of the NAMW.

Table 2.6: RMSE of the GMM-EnKF, EnKF and PF for state estimation in the case with uncertain parameter E_p (case study 3).

Variable	GMM-EnKF	EnKF	PF
$C_M, kgmol/m^3$	0.29	0.26	0.32
$C_I, kgmol/m^3$	0.12	0.10	0.27
T, K	7.2	8.9	10.3
$D_0, kgmol/m^3$	0.111	0.092	0.144
$D_1, kg/m^3$	32.27	35.11	45.34
T_j, K	5.5	5.7	7.5
NAMW	487	869	653

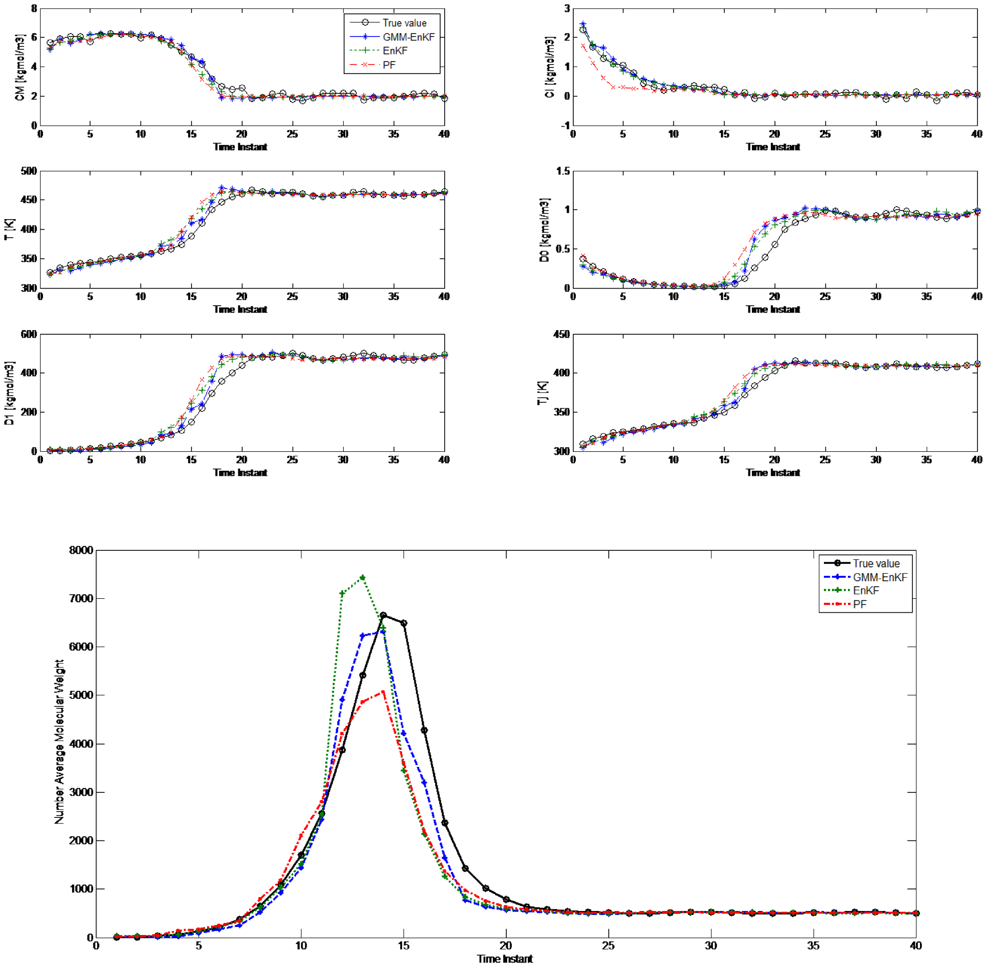


Figure 2.5: Comparison of state estimation with the EnKF-GMM, EnKF and PF for the PMMA process with uncertain parameter E_p (case study 3).

2.4.3.2 State and parameter estimation with uncertain parameter (case study 4)

Next, we compare the performance of the estimators for joint state and parameter estimation. Once again, E_p is the uncertain parameter and its nominal value is kept the same as in case study 3. The parameter E_p is treated as an augmented state for estimation. The prior distribution for E_p has the following characteristics: means of $\mu = [1.9 \times 10^4, 2.5 \times 10^4]$ and covariances of $P = diag(500, 500)$ for its two modes. Bi-modal noise is added to the each particle of the parameter, with means $\mu = [-100, 100]$ and covariances of $P = diag(50, 50)$. Except for the exclusion of process noise, the properties of the simulation are kept the same as in case study 3. Figure 2.6 shows the performance of the estimators in state estimation and Figure 2.7 their performance in estimating the parameter E_p. While the performance of the EnKF in state estimation is comparable to that of the GMM-EnKF, the GMM-EnKF is clearly superior in parameter estimation. The PF has the worst performance among the estimators.

2.4.4 Alternate point estimates for the PF (case study 5)

In the PF, even though the full distribution is obtained, a point estimate for the states is usually obtained by choosing the expectation (mean) of the posterior particles. This is the method we have employed for the PF in the simulations described in the previous sections. However, if the distribution is multimodal, the mean may not necessarily represent the best point estimate, and the mode of the distribution (which is equivalent to the maximum a posteriori estimate) can provide a better estimate [84][6]. We investigate whether this approach can improve the performance of the PF, since we are considering cases where the distributions are multimodal. We apply k-means clustering on the posterior distribution of the particles to identify the modes and the maximum a posteriori estimate with the particle filter, and compare the estimation performance of this PF, called the PF-mode, with the other estimators. The parameters of the simulations are similar to the second case study.

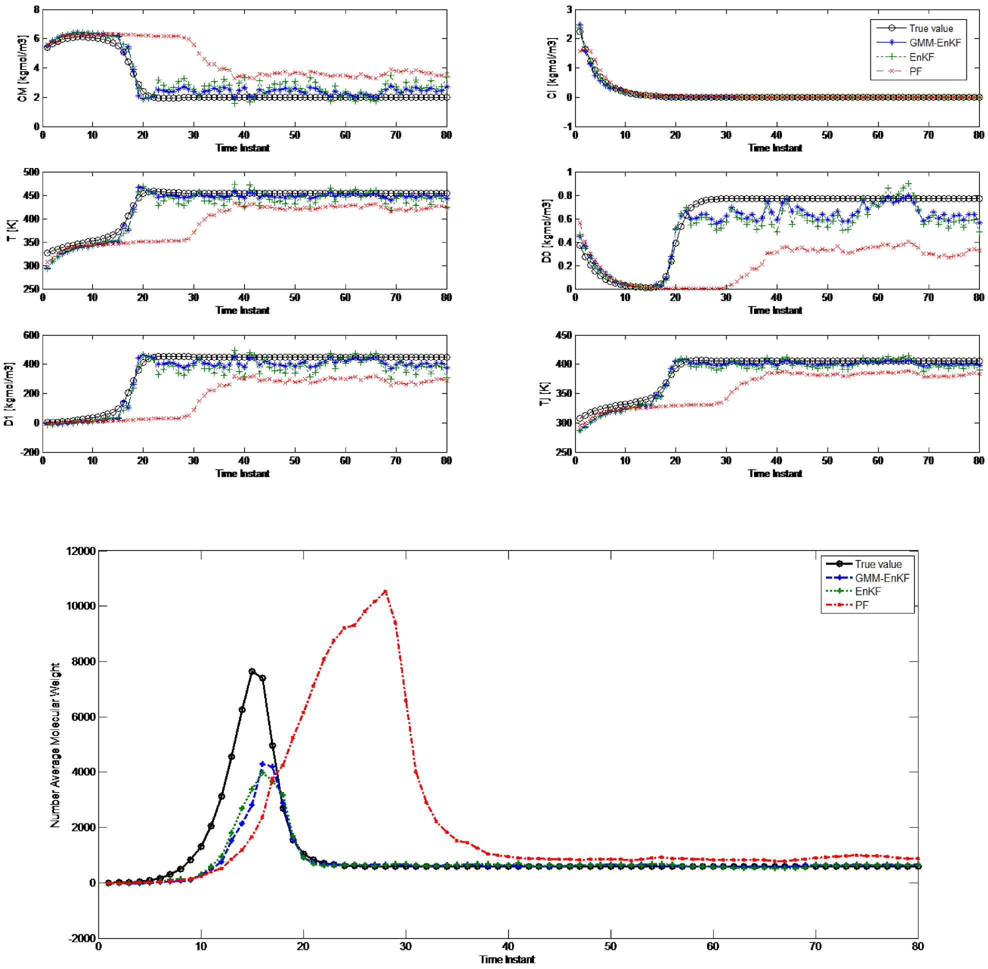


Figure 2.6: Comparison of state estimation with the GMM-EnKF, EnKF and PF for the PMMA process with uncertain parameters (case study 4).

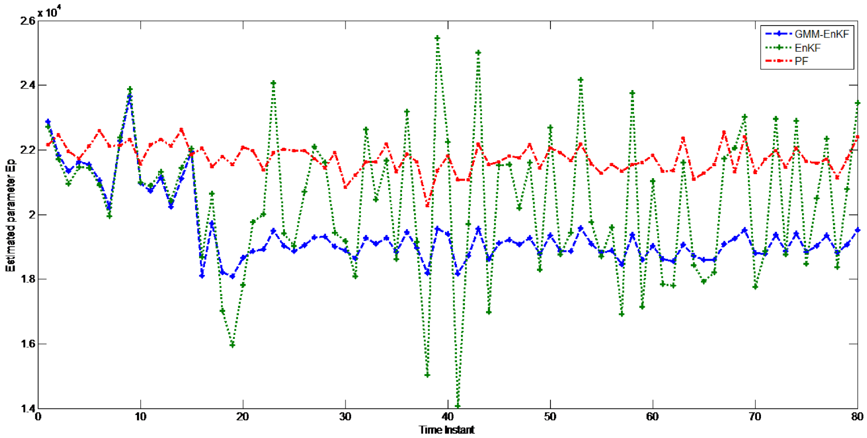


Figure 2.7: Parameter estimation using the GMM-EnKF, EnKF and PF (case study 4).

Figure 2.8 shows the performance of the estimators and the RMSE is described in table 2.7. The PF-mode clearly outperforms the PF and the EnKF; however, the GMM-EnKF has superior performance.

The idea of the PF-mode is very similar to that of the GMM-EnKF. Both of them use clustering to extract modes from the posterior distribution and generate a point estimate based on the information in the modes. However, the GMM-EnKF outperforms the PF-mode because it is more robust to poor initial estimates and model-plant mismatch. Also, if the number of modes in the state distributions varies with time, perhaps even becoming unimodal at some times, using the mode as a point estimate is not necessarily superior to the mean. The GMM-EnKF combines the modes of the distribution in proportion based on the calculated weights to get a point estimate, and can adjust its estimation results in these cases by adjusting the weights of the modes.

Table 2.7: RMSE of the EnKF-GMM, EnKF, PF and PF-mode for state estimation (case study 5).

Variable	GMM-EnKF	EnKF	PF	PF-mode
$C_M, kgmol/m^3$	0.44	0.68	0.68	0.85
$C_I, kgmol/m^3$	0.37	0.14	0.17	0.55
T, K	5.8	11.8	14.4	8.31
$D_0, kgmol/m^3$	0.042	0.062	0.078	0.047
$D_1, kg/m^3$	9.73	36.13	51.38	13.05
T_j, K	5.1	8.2	9.2	7.9
NAMW	559	1400	831	706

2.5 Conclusions

In this chapter, we have proposed an estimator based on a Gaussian mixture model coupled with an ensemble Kalman filter (EnKF-GMM) that is capable of handling multimodal state distributions and demonstrated its performance in simulations on a polymethyl methacrylate process. The EnKF-GMM clearly outperforms the particle filter (PF) and the EnKF in both state and parameter estimation with multimodal distributions. The EnKF is limited by the

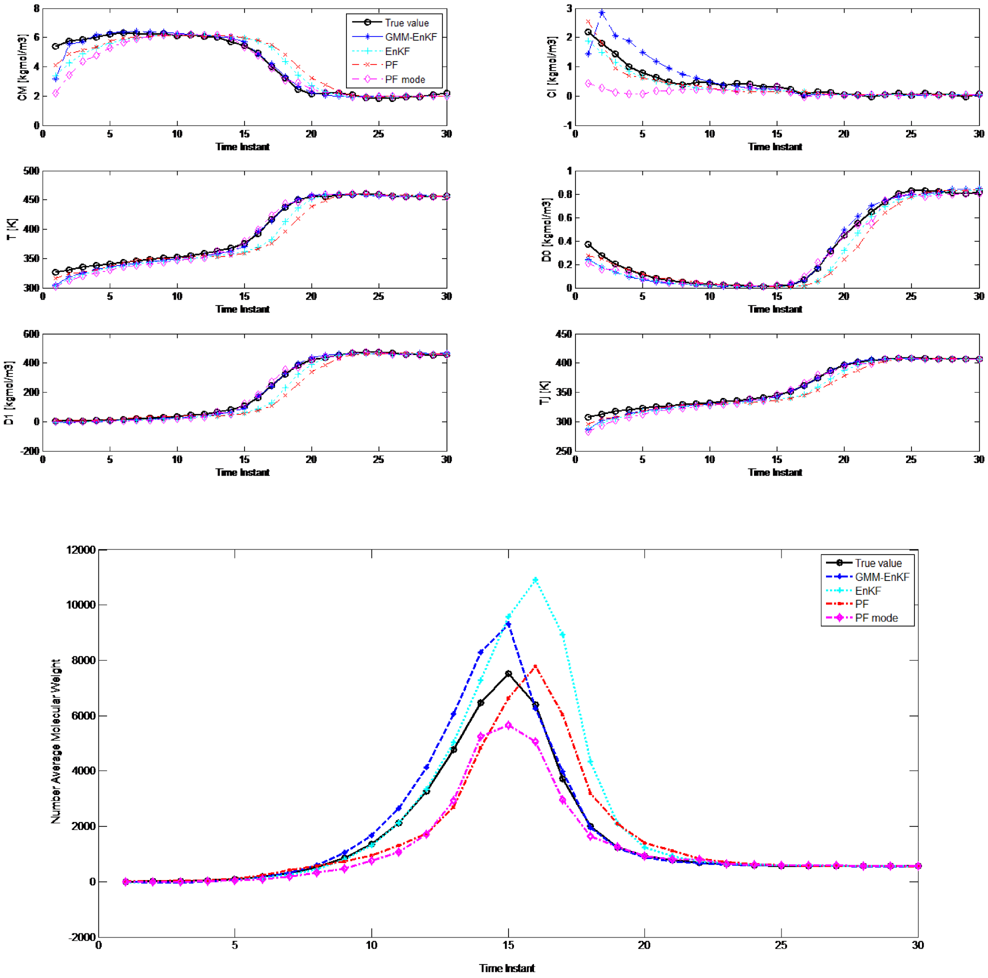


Figure 2.8: Comparison of state estimation with the EnKF-GMM, EnKF, PF and PF-mode (case study 5).

assumption of Gaussian distributions, and the particle filters performance is affected by its lack of robustness with respect to model-plant mismatch. A different choice for obtaining a point estimate with the particle filter, leading to a maximum a posteriori estimate, improves the performance of the PF, but the EnKF-GMM is still superior, indicating that it is the estimator of choice for systems with multimodal state distributions such as polymer processes.

Chapter 3

Constrained extended Kalman filter and ensemble Kalman filter based on Kullback-Leibler (KL) divergence

3.1 Introduction

State estimation plays an important role in achieving good control performance, reliable optimization, and process monitoring. Over the years, various filtering algorithms have been proposed for state estimation, among which the most celebrated Kalman filter (KF) is an optimal filter for linear systems in the presence of Gaussian noise and without constraints. For nonlinear systems, a straightforward extension of the KF is Extended Kalman Filter (EKF), which is based on approximating the nonlinear dynamics of the model with its first-order linearized version to obtain the state estimates [42]. On the other hand, there exist several particle based nonlinear Bayesian filters such as the unscented Kalman filter (UKF), the ensemble Kalman filter (EnKF) and particle filter (PF), which use more sophisticated techniques to handle the nonlinearity of the systems instead of simple linearization. Of these sample based filters, the EnKF has seen highly successful applications on high-dimensional complicated systems, including geophysical systems [102], oceanographical systems [50], land surface models [79], etc. For the UKF, the number of the sigma points required is $2d + 1$, where d is the dimension of the state space. The PF, on the other hand, suffers from the "curse of dimensionality"[91], which implies that the PF requires a large number of particles to avoid the collapse of particles. The EnKF, however, can achieve accurate estimation based on a small number of ensemble for high-dimensional systems. One of the possible reasons is that the EnKF performs a linear update on each of the particles instead of re-weighting the particles, which helps retain the variability of the particles. Regardless of whichever of the above mentioned filters we use, the estimated states might result in physically meaningless values of the states because the physical constraints such as non-negative values of pressure and concentrations are not incorporated in the state estimation procedure [36]. In other words, these recursive filters cannot handle constraints. Therefore, the focus of this work is to develop a constrained state estimation procedure in the recursive framework. In particular, we develop constrained EKF algorithm and constrained EnKF algorithm.

The most common strategy to handle constraints in state estimation is Moving Horizon Estimation (MHE) filter [80]. MHE can naturally handle constraints on states by solving an optimization problem over a finite horizon with constraints easily enforced. In this regard, a moving horizon strategy is proposed in [76] to use a fixed set of measurements to limit the size of the optimization problem being solved at each estimation step. Later, in [77] the MHE procedure is presented for constrained state estimation of discrete-time system, and the moving horizon window requires one to use the approximated arrival cost to account for the past data that are not included in the estimation. However, the choice of the approximation function must ensure that the estimator does not diverge. Despite this, the heavy computation load of the optimization problem and non-recursive nature of the MHE poses difficulties for online applications.

Several methods have been proposed to retain the recursive nature of the Bayesian filters to incorporate constraints. In the EKF framework, [35] proposed two distinct methods to handle inequality constraints. The first method attempts to project the violated posterior estimate back into the constrained space by solving a quadratic programming problem. The second method focuses on restricting the Kalman gain, which produces an updated estimate that lies within the constrained region. In [96], a maximum a *posteriori* (MAP) solution is presented to deal with equality constraints in the constrained EKF framework. Also, an iterative algorithm was proposed to handle inequality constraints.

For the constrained EnKF, in [69], two methods are introduced to incorporate equality constraints on the EnKF. The first is to structure the output equation as a budget constraint. The second is a two-filter approach which first performs a standard EnKF and then formulate another Kalman filter to enforce the equality constraints by considering them as perfect measurements. No state transition model is considered for this second filter. The second filter can be replaced with an optimization problem which solves for the optimal state estimation under constraints. The optimization problem can be structured based on ”least square”, ”maximum likelihood” etc.

Among the recursive methods that incorporate constraints, the most widely used is the Recursive Nonlinear Dynamic Data Reconciliation (RNDDR) approach [99]. The RNDDR attempts to solve an optimization problem by imposing constraints while obtaining the updated state estimates. This is accomplished by embedding the optimization problem into the EKF to form a predictor-corrector framework. However, this approach cannot handle the nonlinearities while calculating the error covariance matrix. Recently, the RNDDR method has been applied in particle based filters, including the EnKF. In [71] and [5], a constrained EnKF framework which formulates the RNDDR on the individual particle was proposed. The key aspect of their work is that the particles which lie outside the constraints are projected back into the constraints. To enhance the effect of the constraints, the initial ensemble is sampled from a truncated distribution to guarantee that each particle satisfies the constraints. Another variant of constrained EnKF formulates the RNDDR optimization only on the mean, and using this estimated mean, the particles in the ensemble have to be adjusted accordingly for next time step. [105] proposed to move the whole unconstrained ensemble by the same distance as the shifting distance of the constrained mean to the unconstrained one, which essentially says moving the ensemble in parallel without changing its original covariance. They argue that the projection operated on individual particle causes the covariance of the ensemble to change significantly from that of the unconstrained ensemble and hence affects the convergence rate of the filter.

Furthermore, the RNDDR approach has been extended to solve constrained particle filter [83], and constrained unscented Kalman Filter [98]. In [83], several variants of RNDDR have been proposed to handle constraints in particle filter. The philosophy of constrained PF is similar to the constrained EnKF. It is important to note that the RNDDR focuses on constraining the mean or the individual particles in the case of sample-based filters, however, the estimation error covariance of these filters is not guaranteed to lie inside the constrained region. Therefore, the main objective of this work is to develop the state estimation strategy that constrains both the mean and error covariance matrix to lie within the constrained region for more accurate state estimates.

In this chapter, we propose a novel Kullback-Leibler (KL) divergence based method to cope with inequality constraints. The KL divergence has been widely used in image retrieval [70], target tracking [34], etc., to measure the difference between two distributions. Note that the idea of Kalman based filters is to propagate and predict the Gaussian distribution at each time step. In this work, we project the unconstrained state distribution obtained using the conventional algorithms into the constrained region by minizing the KL divergence. In other words, we formulate an optimization problem based on the KL divergence to shape a new posterior distribution which satisfies the constraints to approximate the unconstrained one. Compared to the RNDDR, the proposed approach incorporates the constraints by directly finding a replacement distribution which satisfies the constraints instead of indirectly manipulating the mean or the particle of the unconstrained estimation using the RNDDR. We are able to show that the KL divergence based optimization approach can be reduced to the RNDDR under special circumstances. The proposed method is suitable for all Gaussian filters, with minor difference when integrated into the EKF and the EnKF respectively. We are able to show that the constrained EKF and EnKF based on the proposed KL divergence approach outperforms their RNDDR counterparts.

3.2 Preliminaries

In this section, the details of the Extended Kalman Filter and Ensemble Kalman Filter approach for nonlinear state estimation are briefly reviewed. Let $x_k \in R^d$ denote system states and $y_k \in R^m$ denote measurements. Assuming w_k and v_k denote the process and measurement noises with $w_k \sim N(0, Q)$ and $v_k \sim N(0, R)$, a nonlinear discrete time system is given by

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ y_k &= h(x_k) + v_k \end{aligned} \tag{3.1}$$

where $f(.)$ and $h(.)$ are the process and measurement model of the system, respectively.

3.2.1 Extended Kalman filter

The idea of EKF is to linearize the nonlinear model f around $\hat{x}_{k|k}$ to obtain the linear operator A_k of the process model.

$$A_k = \frac{\partial f(x_k, u_k)}{\partial x}\Big|_{\hat{x}_{k|k}, u_k}$$

However, this linearized model is used only in approximating the estimation error covariance $(P_{k+1|k})$, whereas, the nonlinear model is used for predicting the states. Now, the predictor - correction steps of the EKF filter are presented as follows:

Prediction:

At the prediction step, the predicted mean and covariance are calculated as follows:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k) \tag{3.2}$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k \tag{3.3}$$

Update:

The linear measurement operator is calculated as:

$$H_{k+1} = \frac{\partial h(x_k)}{\partial x}\Big|_{\hat{x}_{k+1|k}}$$

The Kalman gain is calculated as:

$$K = P_{k+1|k} H_{k+1} (H_{k+1} P_{k+1|k} H_{k+1}^T + R_k)^{-1} \tag{3.4}$$

The final Kalman update of mean and covariance at time step $k+1$ is given by:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K(y_{k+1} - h(\hat{x}_{k+1|k})) \tag{3.5}$$

$$P_{k+1|k+1} = (I - KH_{k+1})P_{k+1|k} \tag{3.6}$$

It is important to note that the updated state estimation obtained using (3.5) - (3.6) cannot handle constraints. Therefore, the resulting state estimation might yield physically meaningless estimates. Further, [36] has shown that the EKF can fail when the multiple states satisfy the steady state measurements and a poor initial guess of the state is used in the estimator.

3.2.2 The Ensemble Kalman Filter

In this section, we will briefly review the Ensemble Kalman Filter approach for nonlinear state estimation problem [24, 4]. It is a Monte-Carlo based Gaussian filter. The basic idea of EnKF is to approximate the state distribution with an ensemble of randomly sampled particles $\{x_k^i\}_{i=1,2,\cdots,N}$. All the statistical information of the state space can be extracted through this ensemble. At every time instance, each particle is propagated through the model to shape the predicted distribution. Then each particle is updated by assimilating the arriving measurement and the updated statistical information, mean and covariance, can be obtained from the ensemble.

Prediction step:

At time step *k+1* of the prediction step, N particles $x_{k|k}^i, \imath = 1, \cdots, N$ are drawn from the prior distribution $p(x_{k|k})$. The particles $\{x_{k|k}^i\}, i = 1, \cdots, N$ are then propagated through the model to generate a predicted ensemble $\{x_{k+1|k}^i\}_{i=1,\cdots,N}$.

$$x_{k+1|k}^i = f(x_{k|k}^i) + w_k^i \tag{3.7}$$

Also, an ensemble of predicted measurement samples are calculated using the measurement model as follows:

$$y_{k+1|k}^i = h(x_{k+1|k}^i) + v_k^i \tag{3.8}$$

The predicted error cross covariance $P_{k+1|k}^{e,\xi}$ of state and output and the predicted innovation covariance $P_{k+1|k}^{\xi,\xi}$ are not explicitly calculated in the EnKF. Instead they are computed using the predicted ensembles $\{x_{k+1|k}^i\}, \{y_{k+1|k}^i\}_{i=1,\cdots,N}$ described as follows.

$$P_{k+1|k}^{e,\xi} = \frac{1}{N-1} \sum_{i=1}^{N} (e_{k+1|k}^i)(\xi_{k+1|k}^i)^T \tag{3.9}$$

$$P_{k+1|k}^{\xi,\xi} = \frac{1}{N-1} \sum_{i=1}^{N} (\xi_{k+1|k}^i)(\xi_{k+1|k}^i)^T \tag{3.10}$$

$e^i_{k+1|k}$ and $\xi^i_{k+1|k}$ are error matrix defined as:

$$e^i_{k+1|k} = x^i_{k+1|k} - \mu^x_{k+1|k} \tag{3.11}$$

where $\mu^x_{k+1|k} = \frac{1}{N}\sum_{i=1}^{N} x^i_{k+1|k}$.

$$\xi^i_{k+1|k} = y^i_{k+1|k} - \mu^y_{k+1|k} \tag{3.12}$$

where $\mu^y_{k|k-1} = \frac{1}{N}\sum_{i=1}^{N} y^i_{k|k-1}$.

Update step:

The update step updates each of the particle in the predicted ensemble by incorporating the measurement y^{obs}_{k+1}. First, the Kalman gain K is calculated using the two error matrices as follow:

$$K = P^{e,\xi}_{k+1|k}(P^{\xi,\xi}_{k+1|k} + R)^{-1} \tag{3.13}$$

The EnKF then updates each particle in the ensemble as follows:

$$x^i_{k+1|k+1} = x^i_{k+1|k} + K(y^{obs}_{k+1} - y^i_{k+1|k}) \tag{3.14}$$

where y^{obs}_{k+1} is measurement data of time step *k+1*.

The final point estimate provided for time step $k+1$ is given by:

$$\hat{x}_{k+1|k+1} = \frac{1}{N}\sum_{i=1}^{N} x^i_{k+1|k+1} \tag{3.15}$$

The estimated error covariance is given by:

$$P_{k+1|k+1} = \frac{1}{N-1}\sum_{i=1}^{N}(x^i_{k+1|k+1} - \hat{x}_{k+1|k+1})(x^i_{k+1|k+1} - \hat{x}_{k+1|k+1})^T \tag{3.16}$$

For high dimension systems, the direct calculation of the covariance matrix in the traditional Kalman filter can be computationally expensive. The EnKF avoids this problem by operating on a relatively small ensemble instead of explicitly on the covariance itself. This approximation is equivalent to dimension reduction, which enables the EnKF to handle high-dimensional systems. In [106][57], it is found that if the ensemble size exceeds ten, little additional improvement can be found to reduce the propagation error.

3.3 Recursive nonlinear dynamic data reconciliation (RNDDR) approach

The state estimation procedures discussed in the previous section do not impose the constraints on states. Therefore, the state estimates obtained might be physically meaningless. In this section, we briefly review the Recursive Nonlinear Dynamic Data Reconciliation (RNDDR) approach used in constrained state estimation of nonlinear systems [99].

3.3.1 Constrained EKF based on RNDDR approach

The basic idea of RNDDR method is to replace the Kalman update step (Eq. (3.5)) of the EKF algorithm by solving an optimization problem such that the posterior estimates, $\hat{x}_{k+1|k+1}$, are within the constrained region. Given the state constraints of the form, $x_{lb} \leq x \leq x_{ub}$, the constrained state update step requires one to solve the following optimization problem:

$$\begin{aligned}\hat{x}^c_{k+1|k+1} = \underset{x}{argmin}(x - \hat{x}_{k+1|k})^T P^{-1}_{k+1|k}(x - \hat{x}_{k+1|k}) \\ +(y - h(x))^T R^{-1}(y - h(x))\end{aligned} \tag{3.17}$$

$$s.t. \quad x_{lb} \leq x \leq x_{ub}$$

where $P_{k+1|k}$ is the predicted covariance at time step $k+1$, obtained using (3.3).

The above optimization problem provides a better constrained estimate as it penalizes the deviation of both the state vector and the measurement vector. The resulting state estimate will be inside the constrained region in this case. Without the state constraints, the unconstrained optimization problem will reduce to the EKF solution. It is important to note that states are the only decision variables in the optimization problem, and the estimation error covariance matrix is obtained as in conventional EKF using Eqs. (3.4) - (3.6). Therefore, the estimation error covariance might not lie inside the constrained region. In other words, the RNDDR only updates the mean value and the constraint information is not accounted while updating the covariance matrix. Thus, the unconstrained covariance is propagated into subsequent iterations, leading to inaccurate estimation. Other variants of RNNDR approach have been proposed using unscented transformation; interested readers can refer to the works of [97, 98] and [63].

3.3.2 Constrained EnKF based on RNDDR approach

There exist many variants of RNDDR to obtain the constrained state estimation using EnKF. In the first case, the predicted ensemble members, $\{x^i_{k+1|k}\}_{i=1,\cdots,N}$, which are obtained using (3.7), are projected into the constrained region. With constraints added to the states, the RNDDR for constrained EnKF can be formulated as follows:

$$\min_{x^{i,c}_{k+1|k+1}} \; [(y-h(x^{i,c}_{k+1|k+1}))^T R^{-1}(y-h(x^{i,c}_{k+1|k+1})) + (x^{i,c}_{k+1|k+1}-x^i_{k+1|k})^T P^{-1}_{k+1|k} (x^{i,c}_{k+1|k+1}-x^i_{k+1|k})] \tag{3.18}$$

$$s.t. \quad lb \leq x^{i,c}_{k+1|k+1} \leq ub$$

where $x^{i,c}_{k+1|k+1}$ denotes the i^{th} member or particle of the constrained EnKF. Further, solving these optimization problem serves as the update step for the states. On the other hand, the RNDDR can be applied on the unconstrained update members of the ensemble, $x^i_{k+1|k+1}$, and the resulting optimization problem is given by:

$$\underset{x^{i,c}_{k+1|k+1}}{min}\ [(y-h(x^{i,c}_{k+1|k+1}))^T R^{-1}(y-h(x^{i,c}_{k+1|k+1}))+(x^{i,c}_{k+1|k+1}-x^{i}_{k+1|k+1})^T P^{-1}_{k+1|k+1} (x^{i,c}_{k+1|k+1}-x^{i}_{k+1|k+1})] \tag{3.19}$$

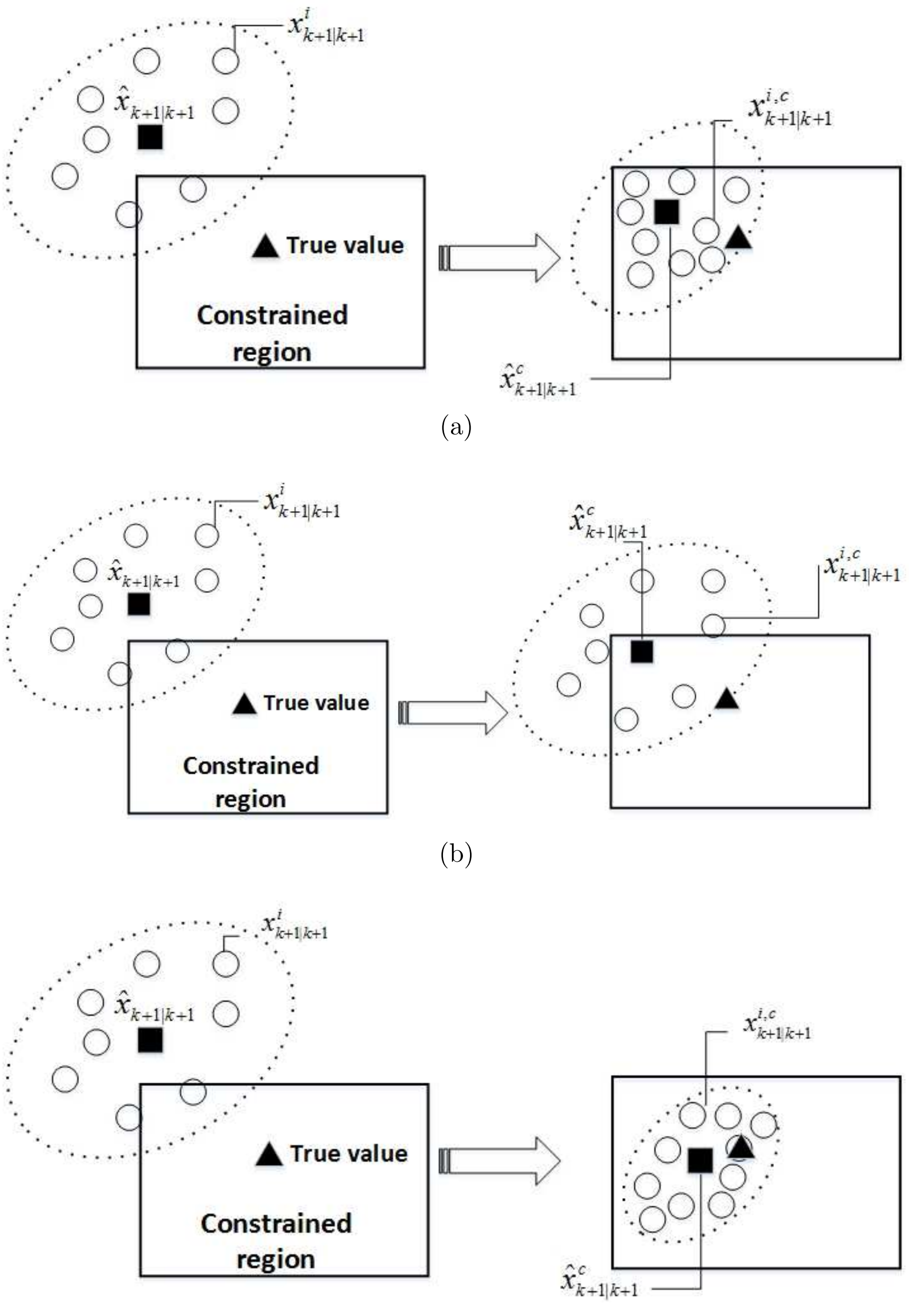
$$s.t.\quad lb \leq x^{i,c}_{k+1|k+1} \leq ub$$

It is important to note that the (3.18) and (3.19) target constraining individual particles of the ensemble. Therefore, N such optimization problems are solved at each time step to enforce constraints in the state estimates. Hence, these two formulations are not computationally attractive. Alternately, the optimization problem can also be formulated in terms of the unconstrained ensemble mean, $\hat{x}_{k+1|k+1}$, as:

$$\underset{\hat{x}^{c}_{k+1|k+1}}{min}\ [(y-h(\hat{x}^{c}_{k+1|k+1}))^T R^{-1}(y-h(\hat{x}^{c}_{k+1|k+1}))+(\hat{x}^{c}_{k+1|k+1}-\hat{x}_{k+1|k+1})^T P^{-1}_{k+1|k+1}(\hat{x}^{c}_{k+1|k+1}-\hat{x}_{k+1|k+1})] \tag{3.20}$$

$$s.t.\quad lb \leq \hat{x}^{c}_{k+1|k+1} \leq ub$$

All above three constrained formulations penalize both the deviation of the constrained estimate from the unconstrained one and the measurement error. However, constraining only the mean value is much more computationally efficient than constraining individual particle since (3.20) solves only one optimization problem at every time step in contrast to solving N problems in the individual member formulation. The basic idea of these approaches can be illustrated as figure (a) and (b) in Figure 3.1. It is important to note that none of the above formulation accounts for the constraints in the estimation of variance. In other words, the update of constrained covariance $P^c_{k+1|k+1}$ remains the same as $P_{k+1|k+1}$ in (3.16). As is shown in the EKF case, unconstrained covariance would affect the convergence rate significantly. A crucial step after the obtaining the constrained mean is to adjust the particles accordingly since they are the ones that pass the constraints information to the next iteration. Yang et



(a)

(b)

(c)

Figure 3.1: Illustration of (a) RNDDR on unconstrained particles; (b) RNDDR on unconstrained mean; and (c) Proposed KL divergence based EnKF

al. [105] proposes to shift all the particles the same distance as that of constrained mean. To this end, we define the distance, δ_{k+1}, as:

$$\delta_{k+1} = |\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1}| \tag{3.21}$$

Correspondingly, all the particles are moved by the distance δ as is given as follows.

$$x^{i,c}_{k+1|k+1} = x^i_{k+1|k+1} + \delta_{k+1} \tag{3.22}$$

This particle adjustment technique maintains the unchanged covariance of the ensemble, which reduces the negative impact on the convergence rate of the original EnKF. However, a large portion of the particles might fall out of the constrained region if the constrained region is narrower than the covariance of the ensemble, which might result in inaccurate state estimates as well. In other words, although (3.18) and (3.19) guarantee that every particle lies within the constraints, the poor initial guess of the unconstrained ensemble tends to have the particles sit close to the boundary to compensate for the cost $(x^{i,c}_{k+1|k+1} - x^i_{k+1|k+1})^T P^{-1}_{k+1|k+1}(x^{i,c}_{k+1|k+1} - x^i_{k+1|k+1})$ in the RNDDR framework. This happens especially when the measurement noise R is large. Because in this case the correction of states is poor due to unreliable measurement, the measurement term of the cost function can be ignored. The accumulation of particles around the boundary might cause the variability of particles in the ensemble to reduce significantly. Although the EnKF does not collapse as easily as the PF for the loss of the variability, it is still a loss of information.

3.4 Proposed KL divergence based constrained state estimation

3.4.1 Motivation for using KL divergence

Owing to the assumption that the states x_k follow Gaussian distribution at the prediction step (i.e., $\hat{x}_{k+1|k} \sim N(\mu^x_{k+1|k}, P_{k+1|k})$), and at the correction step (i.e., $\hat{x}_{k+1|k+1} \sim N(\mu^x_{k+1|k+1}, P_{k+1|k+1})$), we can project the unconstrained state estimation results, obtained using the conventional procedure presented in Section 3.2, into the constrained region. To this end, we propose to use the KL divergence measure. It measures the distance between two arbitrary probability distributions f(x) and g(x), and it is defined as:

$$D(f||g) = \int f(x) log \frac{f(x)}{g(x)} dx \tag{3.23}$$

In general, filtering procedure introduced in the previous section gives the estimation of posterior distribution $p(x_k|y_{1:k})$ at time step k. $p(x_k|y_{1:k})$ could be any arbitrary distribution for nonlinear systems. Our goal is to approximate $p(x_k|y_{1:k})$ with another distribution $q(x_k)$, such that the constraints are satisfied. This approximated distribution $q(x_k)$ is determined by minimizing the KL divergence from $p(x_k|y_{1:k})$ to minimize the loss of information obtained from the unconstrained filtering step. However, the KL divergence is not a true distance measure because it has the following properties:

1. KLD can never be negative, $D(f(x)||g(x)) \geq 0$. $D(f(x)||g(x)) = 0$ if and only if $f(x) = g(x)$ for every x of the support.

2. KLD is not symmetric, i.e. $D(f||g) \neq D(g||f)$ where $D(f||g)$ denotes forward KLD and $D(g||f)$ denotes reverse KLD.

3. KLD does not satisfy the triangle inequality of a true distance measure, i.e. $D(f||g) + D(g||h) \geq D(f||h)$.

Therefore, using the KLD in different directions results in different $g(x)$. However, it has been widely used to find an approximated distribution $g(x)$ of the true distribution $f(x)$ in optimization framework. Further, if $f(x)$ and $g(x)$ are of both Gaussian distributions, the KL divergence has a closed form which can be expressed as:

$$\begin{aligned} D(f||g) = \frac{1}{2}[log\frac{|\Sigma_g|}{|\Sigma_f|} + Tr(\Sigma_g^{-1}\Sigma_f) - d \\ +(\mu_f - \mu_g)^T\Sigma_g^{-1}(\mu_f - \mu_g)] \end{aligned} \tag{3.24}$$

where $f(x) = N(x; \mu_f, \Sigma_f)$, $g(x) = N(x; \mu_g, \Sigma_g)$ and d is the dimension of x.

Illustration 1:

The objective of this example is to demonstrate the approximation of a distribution obtained by minimizing forward and reverse KL divergence. For this purpose, let us assume $f(x) \sim N(x; \mu_f, \sigma_f^2)$, where $\mu_f = 10, \sigma_f = 3$. An inequality constraint $8 < x < 15$ is put on f(x). The constrained distribution $g(x) \sim N(x; \mu_g, \sigma_g^2)$ is obtained by minimizing the forward or backward KLD. The optimization problem is formulated to guarantee g(x) lies in the constrained region $8 < x < 15$ with 99% confidence. Figure 3.2 shows the distribution of g(x) using the forward KLD and reverse KLD respectively. In the figure, The region within the two dotted lines is the constrained region. We can see that the g(x) of both methods are very close to each other. When the constrained region is located close to the mean of f(x), similar g(x) is generated with both methods.

$$\begin{aligned} &\min_{\mu_g,\sigma_g} D(f||g) \quad or \quad D(g||f) \\ s.t. \quad & 8 < \mu_g < 15 \\ & 8 < \mu_g + 3\sigma_g < 15 \\ & 8 < \mu_g - 3\sigma_g < 15 \end{aligned}$$

Figure 3.3 shows the constrained g(x) using the forward KLD and reverse KLD respec-

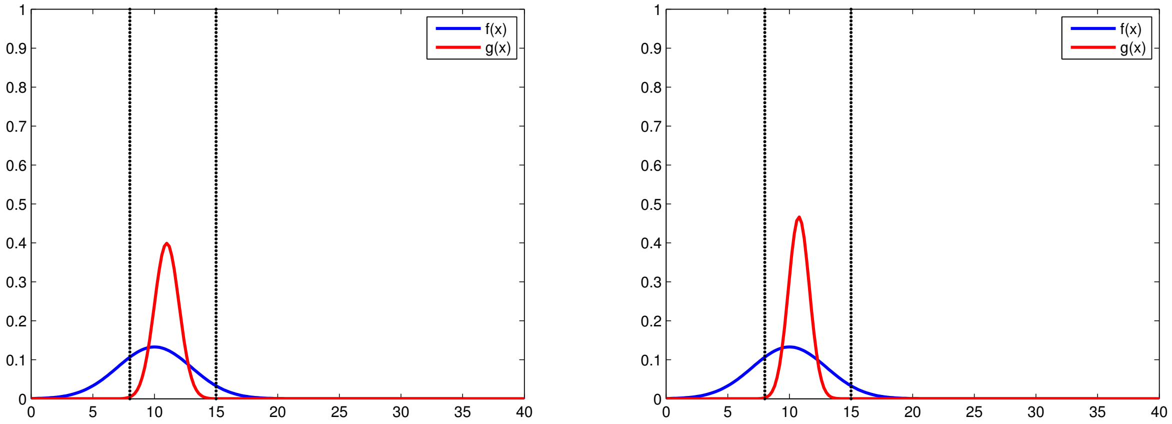


Figure 3.2: Results of $g(x)$ using forward KLD(left) and reverse KLD(right) with constraint $8 < x < 15$

tively when we change the inequality constraint to $11 < x < 20$. Compared to the previous case, the resulted g(x) of the two methods is noticeable different. Such difference occurs because the constrained region [11 20] does not include the peak of f(x),in other words, it is located to the tail of f(x). As is introduced earlier, the mode-seeking behavior of the reverse KLD drives g(x) close to the mean of f(x) and meanwhile sets most part of g(x) to zero where f(x) is close to zero, hence the sharp peak. The peaks will get even sharper if the constrained region keeps moving to the tail of f(x). In conclusion, the forward KLD produces a g(x) which stretches all over the constrained region wherever the constrained region is located, while reverse KLD produces a g(x) which attempts to get as close to the mean of f(x) as possible. If the primary concern is to achieve minimum modification of the unconstrained point estimates, i.e. the mean value, the g(x) from the reverse KLD is a preferable option in the presence of the constraints, although it might cause the covariance to shrink significantly.

In this work, the reverse KLD will be used as the cost function in the optimization problem because it has less computation cost, and it yields an estimate closer to the unconstrained estimate. Further, as we will show in subsequent section that the resulting optimization problem can be cast as convex problem which can be solved to global optimality.

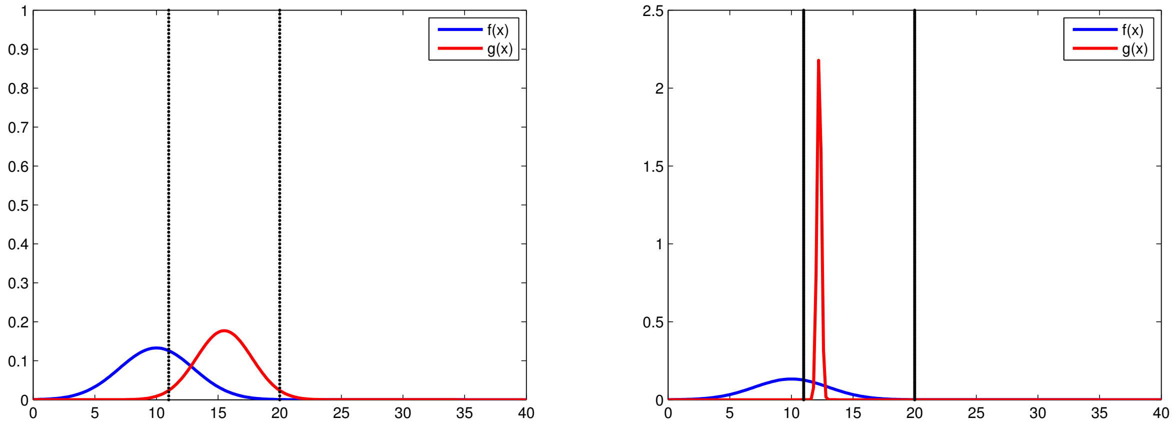


Figure 3.3: Results of $g(x)$ using forward KLD(left) and reverse KLD(right) with constraint $11 < x < 20$

3.4.2 Optimization formulation for KL based constraining approach

The motivation of our proposed method is to embrace the constraints in updating both the states and estimation error covariance matrix. Recall that the update equations of unconstrained state estimates ((3.5) - (3.6) for the case of EKF, and (3.5) - (3.6) for the case of EnKF), represent the ellipsoidal uncertain region around the state estimate ($\hat{x}_{k+1|k+1}$). The main idea of our approach is to project the unconstrained solution into the constrained region such that both the state and its estimation error lie within the constrained space. Since Kullback-Leibler (KL) divergence is known to be a measure of similarity between two distributions, we seek to determine the multivariate Gaussian distribution in the constrained region that is close and similar to the unconstrained multivariate Gaussian distribution obtained from state estimation procedures presented in Section 3.2.

Let us denote the unknown constrained distribution $g(x) = N(x; \hat{x}^c_{k+1|k+1}, P^c_{k+1|k+1})$ obtained by projecting the unconstrained distribution $f(x) = N(x; \hat{x}_{k+1|k+1}, P_{k+1|k+1})$. Recall that $\hat{x}_{k+1|k+1}$ and $P_{k+1|k+1}$ are the solution of the unconstrained Kalman update equations (3.5) -(3.6). Since we are considering the complete distribution information of the unconstrained EKF solution to be projected into the constrained region, we seek the state covariance to be completely inside the constrained region. It is important to notice that the state covariance $P_{k+1|k+1}$ obtained from the EKF signifies the ellipsoid representation around the state estimates. Therefore, to constrain the state covariance, we utilize the following expression of ellipsoid:

$$\mathcal{E} = \{\hat{x}_{k+1|k+1} + \alpha S_{k+1} z \mid \|z\|_2 \leq 1\} \tag{3.25}$$

where $\hat{x}_{k+1|k+1}$ is the center of the ellipsoid, S_{k+1} is the positive square root of $P_{k+1|k+1}$ and α depends on the confidence limit (e.g., $\alpha = 2$ signifies a confidence limit of 95%). In order to bound the state covariances, we enforce the following constraints:

$$\mathcal{E} = \{(x_{lb} \leq \hat{x}_{k+1|k+1} + \alpha S_{k+1} z \leq x_{ub}) \mid \|z\|_2 \leq 1\} \tag{3.26}$$

or equivalently, the above constraint can rewritten as:

$$\tilde{x} := \hat{x}_{k+1|k+1} + \alpha S_{k+1} z \mid \|z\|_2 \leq 1 \tag{3.27}$$

$$h_i^T \tilde{x} + t_i \leq 0; i = 1, \ldots, m \tag{3.28}$$

where h_i is the i^{th} row of the matrix $H = [I; -I]$ and t_i is the i^{th} element of vector $t = [x_{ub}; -x_{lb}]$. Now, the optimization formulation that can simultaneously determine the state updates and state covariances of the constrained filtering problem can be formulated as follows:

$$\min_{\hat{x}^c_{k+1|k+1}, P^c_{k+1|k+1}} \quad D(g||f) \tag{3.29}$$

$$\text{s.t.} \quad x_{lb} \leq \hat{x}^c_{k+1|k+1} \leq x_{ub} \tag{3.30}$$

$$S_{k+1} = P^{c1/2}_{k+1|k+1} \tag{3.31}$$

$$\tilde{x} := \hat{x^c}_{k+1|k+1} + \alpha S_{k+1} z \mid \|z\|_2 \leq 1 \tag{3.32}$$

$$h_i^T \tilde{x} + t_i \leq 0; i = 1, \ldots, m \tag{3.33}$$

The above optimization problem is a semi-infinite optimization problem and it is not computationally tractable owing to the nonlinear matrix constraint (3.31) and infinite dimensional constraints (3.32). Further, the objective function is non-convex.

3.4.3 Convex reformulations

In this subsection, we present the convex optimization techniques to reformulate the above infinite dimensional optimization problem such that it can be cast as a convex optimization problem. To this end, first let us introduce some of the definitions that will enable us to reformulate the problem. For more details on the definitions, readers are referred to [12].

Definition 3.1: The epigraph of a function $f : R^n \rightarrow R$ is defined as [12]:

$$\textbf{epi f} = \{(x,t)|x \in \textbf{dom}f, f(x) \leq t\}$$

The epigraph of a function is the region above the function. For a convex function f, the minimization of f is equivalent to the finding the lowest point of the epigraph.

Definition 3.2: For any symmetric matrix M of the form:

$$M = \begin{bmatrix} U & V \\ V^T & W \end{bmatrix}$$

the Schur complement of W is defined as $U - VW^{-1}V^T$. If W is invertible, then we have:

If $W \succ 0$, then $M \succeq 0$ if $U - VW^{-1}V^T \succeq 0$.

Theorem 3.1: The set of linear constraints $h_i^T z \leq t_i, z \in \mathcal{E}$ can be expressed as a set of second order cone constraints of the form ${h_i}^T z_{ss} + \alpha \|Sh_i\|_2 \leq t_i$.

Proof[12]: Recall that $z := z_{ss} + \alpha S\tilde{z}, ||\tilde{z}|| \leq 1$. Consider the infinite dimensional constraint:

$$\begin{aligned}
&h_i^T z \leq t_i \; \forall z \in \mathcal{E} \\
&\iff \sup\{h_i^T z \mid z \in \mathcal{E}\} \leq t_i \\
&\iff \sup\{h_i^T (z_{ss} + \alpha S\tilde{z}) \mid \|z\|_2 \leq 1\} \leq t_i \\
&\iff h_i^T z_{ss} + \sup\{h_i^T \alpha S z \mid \|z\|_2 \leq 1\} \leq t_i \\
&\iff h_i^T z_{ss} + \alpha\|Sh_i\|_2 \leq t_i
\end{aligned}$$

Now, let us consider the objective function of the proposed optimization problem (3.29) which can be expressed using the closed form of the KLD for Gaussian distributions.

$$\begin{aligned}
D(g||f) = \frac{1}{2}[log\frac{|P_{k+1|k+1}|}{|P^c_{k+1|k+1}|} + Tr(P^{-1}_{k+1|k+1}P^c_{k+1|k+1}) - d \\
+(\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})^T P^{-1}_{k+1|k+1}(\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})]
\end{aligned} \tag{3.34}$$

Substituting $P^c{}_{k+1|k+1}$ with its matrix square root, $P^c{}_{k+1|k+1} = S_{k+1}S^T_{k+1}$, and using the properties of determinant $(det(AB) = det(A)det(B); det(A) = det(A^T))$, Equation (3.34) can be written as:

$$\begin{aligned}
D(g||f) = \frac{1}{2}[log\mathbf{det}(P_{k+1|k+1}) - 2log\mathbf{det}(S_{k+1}) + Tr(P^{-1}_{k+1|k+1}S_{k+1}S^T_{k+1}) - d \\
+(\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})^T P^{-1}_{k+1|k+1}(\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})]
\end{aligned} \tag{3.35}$$

Using the property that trace operator is invariant under cyclic permutations (i.e., $Tr(UVW) = Tr(VWU) = Tr(WUV) \neq Tr(VUW)$), we can rewrite the trace term as:

$$Tr(P^{-1}_{k+1|k+1}S_{k+1}S^T_{k+1}) = Tr(S^T_{k+1}P^{-1}_{k+1|k+1}S_{k+1}) \tag{3.36}$$

Substituting Eq.(3.36) into Eq.(3.35), we have:

$$D(g||f) = \frac{1}{2}[log\mathbf{det}(P_{k+1|k+1}) - d - 2log\mathbf{det}(S_{k+1}) + Tr(S_{k+1}^T P_{k+1|k+1}^{-1} S_{k+1})$$
$$+(\hat{x}_{k+1|k+1}^c - \hat{x}_{k+1|k+1})^T P_{k+1|k+1}^{-1}(\hat{x}_{k+1|k+1}^c - \hat{x}_{k+1|k+1})] \qquad (3.37)$$

The first two terms in Eq.(3.37) are constant and it is well known that the negative log determinant function of a positive definite matrix X, $f(x) = log\mathbf{det}(X)$, is convex, hence $log\mathbf{det}(S_{k+1})$ is convex. However, the trace term is non-convex because the decision variables are expressed in quadratic form of unknown matrices. Introducing auxiliary variables Y and q, we can use the definition of epigraph to express the trace term as:

$$\min_{Y,q} q \qquad (3.38)$$
$$s.t. \quad Tr(Y) - q \leq 0$$
$$Y - S_{k+1}^T P_{k+1|k+1}^{-1} S_{k+1} \succeq 0 \qquad (3.39)$$

Note that Eq.(3.39) should be $Y = S_{k+1}^T P_{k+1|k+1}^{-1} S_{k+1}$. The matrix equality is strict with nonlinear term and hence difficult to optimize. Using the definition of Schur complement, the matrix inequality in Eq.(3.39) can be written in linear matrix inequality(LMI) form:

$$\begin{bmatrix} Y & S_{k+1} \\ S_{k+1}^T & P_{k+1|k+1} \end{bmatrix} \succeq 0 \qquad (3.40)$$

These convex optimization tricks enable one to replace the trace term with a linear term, q, in the objective function along with an upper bound for the trace term, and an LMI constraint.

Now, let us consider reformulating the infinite dimensional constraints (3.32) - (3.33). These constraints can be rewritten in terms of the following second order cone constraints

[12]:

$$\alpha||S_{k+1}h_i||+h_i^T\hat{x}_{k+1|k+1} \leq t_i \tag{3.41}$$

The resulting optimization problem for updating the state estimates and its covariance is presented below:

$$\begin{aligned}
\min_{\hat{x}^c_{k+1|k+1}, S_{k+1}} \frac{1}{2}[log\mathbf{det}(P_{k+1|k+1}) - d - 2log\mathbf{det}(S_{k+1}) + q \\
+ (\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})^T P^{-1}_{k+1|k+1}(\hat{x}^c_{k+1|k+1} - \hat{x}_{k+1|k+1})] \quad (3.42)\\
\text{s.t.} \quad x_{lb} \leq \hat{x}^c_{k+1|k+1} \leq x_{ub} \quad (3.43)\\
Tr(Y) \leq q \quad (3.44)\\
\begin{bmatrix} Y & S_{k+1} \\ S^T_{k+1} & P_{k+1|k+1} \end{bmatrix} \succeq 0 \quad (3.45)\\
\alpha||S_{k+1}h_i||+h_i^T\hat{x}_{k+1|k+1} \leq t_i \quad (3.46)
\end{aligned}$$

In the above formulation, the decision variables are $\hat{x}^c_{k+1|k+1}$, S_{k+1}, q, and Y. The solution to this optimization problem directly yields the state updates, $\hat{x}^c_{k+1|k+1}$, whereas the error covariance of the states ($P^c_{k+1|k+1}$) can be updated from S_{k+1} as:

$$P^c_{k+1|k+1} = S_{k+1}S^T_{k+1} \tag{3.47}$$

After acquiring the constrained estimate $\hat{x}^c_{k+1|k+1}$ and $P^c_{k+1|k+1}$, pass them to the next time step as prior estimates. It should be noted that the proposed optimization problem is convex and hence, it can be solved for global optimality using available convex optimization tool, CVX, which is a MATLAB based software for solving convex optimization problems [31]. Further, the proposed constraining step is applicable to any Gaussian filter. In Table 3.1, we present the constrained EKF algorithm that uses the proposed optimization problem to constrain the state estimates. Also, this optimization formulation can be used in the EnKF

Table 3.1: Constrained EKF algorithm based on KL divergence

1	Given the initial values for $x_{0\|0}$ and $P_{0\|0}$
2	At time step k, obtain the unconstrained EKF solution, $\hat{x}_{k+1\|k+1}$ and $P_{k+1\|k+1}$, using (3.5) and (3.6), respectively
3	Solve the optimization problem [(3.42) to (3.46)] using CVX to obtain the constrained EKF solution, $\hat{x}^c_{k+1\|k+1}$ and $P^c_{k+1\|k+1}$.
4	Use the constrained EKF solution obtained in Step 3 to do prediction at time step $k+1$ using (3.2) - (3.3), and proceed to Step 2.

with suitable redistribution of particles.

3.4.4 Re-distribution or regeneration of particles for constrained EnKF

At every time step, the constrained EKF ends with the obtained constrained mean $\hat{x}^c_{k+1|k+1}$ and constrained covariance $P^c_{k+1|k+1}$ using the KLD based approach. This is because the mean and covariance are the variables passed to the next step in the recursive framework of the EKF. However, this is not the case for the EnKF for the reason that the EnKF relies on the particles in the ensemble to pass the information to the next step. Therefore, at every iteration, not only do we have to further adjust the updated mean and covariance by solving KLD based optimization approach, we also have to adjust the updated particles $\hat{x}^i_{k+1|k+1}$ according to the constrained mean and covariance, such that the information of the constraints can be accounted for in the next iteration. Both the change of mean and covariance compared to the unconstrained values caused by the constraining step should be accounted for when re-distributing particles. The unconstrained state $x_{k+1|k+1}$ is transformed to the constrained state $x^c_{k+1|k+1}$ as follows:

$$x^c_{k+1|k+1} = W x_{k+1|k+1} + z \tag{3.48}$$

where $x_{k+1|k+1}$ is a $d \times 1$ vector; W is a $d \times d$ diagonal matrix; z is a $d \times 1$ vector.

Assuming the state variables are independent of each other, i.e. the state space is orthogonal, the particles can be re-distributed separately on each dimension. The change of the mean that is stored in vector z calculates the moving distance of each particle on each dimension l of the state space. The change of the standard derivation in Eq.(3.49) stored in matrix W quantifies how much the cluster shrinks or expands on each dimension. Since the state space is independent, W is a diagonal matrix. Matrix W and z are calculated as follows:

$$W^2(l,l) = P^c_{k+1|k+1}(l,l)/P_{k+1|k+1}(l,l) \quad l = 1, \cdots, d \tag{3.49}$$

$$z(l) = \hat{x}^c_{k+1|k+1}(l) - W(l,l)\hat{x}_{k+1|k+1}(l) \quad l = 1, \cdots, d \tag{3.50}$$

where d is the dimension of the state x.

With matrix W and z, the constrained particle $x^{i,c}_{k+1|k+1}$ can be calculated as follows:

$$x^{i,c}_{k+1|k+1}(l) = W(l,l)x^i_{k+1|k+1}(l) + z(l) \quad l = 1, \cdots, d \tag{3.51}$$

If the correlation of the state variables cannot be ignored, the problem will be much more complicated because the re-distribution of particles cannot be performed per dimension. In this case, matrix W is not diagonal. The mean and covariance of the constrained variable $x^c_{k+1|k+1}$ can be written in terms of those of $x_{k+1|k+1}$ as follows.

$$\hat{x}^c_{k+1|k+1} = E[x^c_{k+1|k+1}] = W\hat{x}_{k+1|k+1} + z \tag{3.52}$$

$$P^c_{k+1|k+1} = Cov[x^c_{k+1|k+1}] = WP_{k+1|k+1}W^T \tag{3.53}$$

We solve Eq.(3.52) and Eq.(3.53) for each element in matrix W and z. However, Eq.(3.53) is a under-determined equation. This is because we are calculating the values of all the elements of W, the degree of freedom of W is at least $\frac{d(d+1)}{2}$, which is the number of elements of W considering W is symmetric. The number of constraints given to solve these elements

Table 3.2: Constrained EnKF algorithm based on KL divergence

1	Generate initial particles $\{x_0^i\}, i = 1, \dots, N$ based on the initial guess $\hat{x}_0$ and P_0.
2	At time step $k+1$, obtain the unconstrained ensemble $x_{k+1\|k}^i$ using Eq.(3.14), and the unconstrained solution $\hat{x}_{k+1\|k+1}$ and $P_{k+1\|k+1}$, using (3.15) and (3.16), respectively.
3	Solve the optimization problem [(3.42) to (3.46)] using CVX to obtain the constrained EnKF solution, $\hat{x}_{k+1\|k+1}^c$ and $P_{k+1\|k+1}^c$.
4	Redistribute the constrained particle $x_{k+1\|k+1}^{c,i}$ using Eq.(3.51) or regenerate the constrained ensemble based on $\hat{x}_{k+1\|k+1}^c$ and $P_{k+1\|k+1}^c$.
5	Go to step 2

is d, which is the dimension of the covariance $P_{k+1|k+1}^c$ and $P_{k+1|k+1}$.

$$\frac{d(d+1)}{2} - d = \frac{d(d-1)}{2} \geq 0, \quad when \quad d \geq 1 \tag{3.54}$$

Eq.(3.54) indicates that there are fewer equations than the unknown variables when $d > 1$. Therefore, Eq.(3.53) cannot yield a unique solution of matrix W.

Regenerate the particles

Therefore, in the case of severely correlated states, the alternative approach to turn to is to regenerate the whole ensemble of particles from the constrained Gaussian distribution at the end of every time step, i.e. $x_{k+1|k+1}^{i,c} \sim N(x; \hat{x}_{k+1|k+1}, P_{k+1|k+1}^c), i = 1, \cdots, N$. Note that although the ensemble produced by the regeneration approach is similar to the one produced by the previous re-distribution approach, it has the risk of losing information gained from the filtering step or adding untrue information to the ensemble. This is because it only relies on the two moments to generate the new ensemble, while the the previous re-distribution approach makes adjustments based on individual particles. The proposed constrained EnKF algorithm is presented in Table 3.2.

3.5 Simulation

3.5.1 Case study: A two-state batch reaction system

To demonstrate our proposed approach, let us consider the batch reaction system studied by: [71, 83] and [96]. Consider a gas phase reaction given by:

$$2A \to B \quad k = 0.16 \tag{3.55}$$

Let the states of the process be partial pressures of A and B, $x = [P_A, P_B]$, and the total pressure $P = P_A + P_B$ is measured. Assuming the reaction occurs in a well-mixed isothermal batch reactor, the state space model can be written as:

$$\dot{x} = f(x) = \begin{bmatrix} -2 \\ 1 \end{bmatrix} kP_A^2, \quad y = \begin{bmatrix} 1 & 1 \end{bmatrix} x \tag{3.56}$$

The model is discretized at an interval of 0.1 s and simulated from an initial value of $x_0 = [3, 1]$. The process and measurement noises are both assumed to be Gaussian with zero mean. Their respective covariances are $Q = diag(0.001^2, 0.001^2)$ and $R = 0.1^2$. A poor initial guess is given to the state estimator with $\bar{x}_0 = [0.1, 4.5]$ and a large covariance matrix $P_0 = diag(6^2, 6^2)$ is used.

The proposed constrained EKF and EnKF are applied to this case study respectively.

Constrained EKF:

First the EKF without any constraint is applied on the model to estimate the states $[P_A, P_B]$. It can be seen from Figure 3.4 that the estimation results using the unconstrained EKF diverge from the true values. This poor estimation performance is caused because of the intentionally chosen large initial error. Then the constrained estimation with EKF is performed with the following inequality constraints imposed on state variables:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} P_A \\ P_B \end{bmatrix} \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

Figure 3.5 shows the comparison of constrained estimation results using the RNDDR method and our proposed KL divergence based method. Both methods yield convergent estimation results because of the incorporation of constraints, and they both perform well in light of satisfying constraints. The reason for the superior performance of KL based method is that RNDDR only enforces the mean value to be inside the constraints and does not constrain the state error covariance matrix, whereas, the KL based method adjusts both the mean and error covariance by incorporating constraints. The modified covariance is directly propagated on to the calculation of Kalman gain in the next iteration, providing a faster impact on the estimation from the constraints compared to the RNDDR solution.

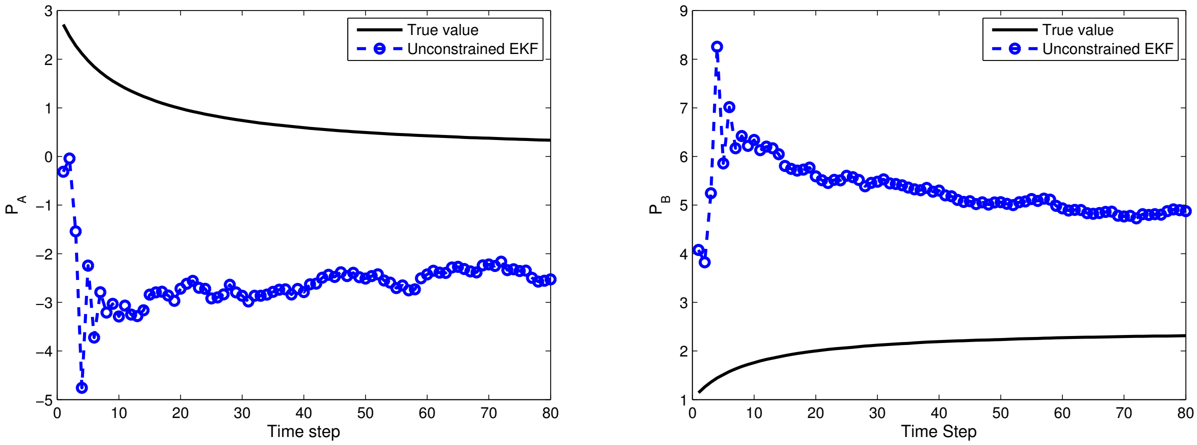


Figure 3.4: State estimates using unconstrained EKF approach

In Figure 3.6, we present the state estimate and its error covariance at first and second time steps to show that the constrained error covariance is obtained using KL method. The ellipses represent the two-dimensional projection of the state PDF on xy plane and the rectangular region denotes the constrained space. It should be noted that, at first

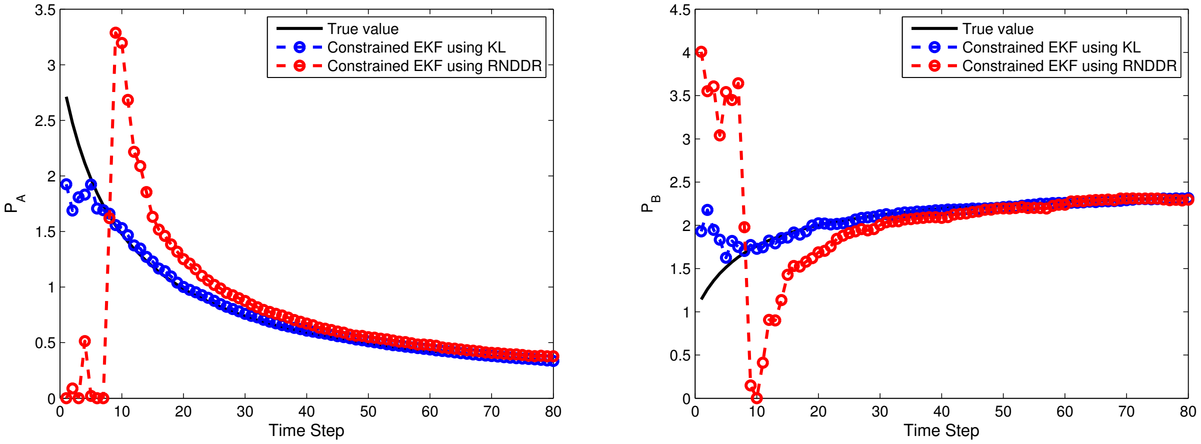


Figure 3.5: Comparison of state estimates obtained using the proposed KL approach with the RNDDR approach

time step, the RNDDR does not deviate much from the unconstrained result because it only projects the violated mean $\hat{x} = [-0.16 \quad 4.23]$ to the boundary (i.e., the RNDDR estimate is $\hat{x} = [0.001 \quad 4.25]$), whereas, the KL method shrinks the original distribution by a large portion to fit inside the constrained region. At second time step, the estimation error covariance obtained using the KL method shrinks even further into the constrained region, whereas, the RNDDR still yields a large error covariance.

Figure 3.7 shows the evolution of variance of P_A for first 20 time steps. P_B also produces a very similar results. The KL method provides a much smaller variance from the beginning, while the RNDDR is stuck at a large error for several time steps. The smaller covariance obtained from the KL method remedies the large error covariance exerted on the initial state distribution since the prediction part of the Kalman update produces more reliable results with smaller covariance. The KL based method has a faster convergence rate, and also results in smaller estimation error. Table 3.3 presents the root mean square error (RMSE) of estimated states, P_A and P_B, and computation time using the RNDDR method and our proposed KL based method, respectively. The higher computational time of the KL method can be attributed to solving conic optimization problems for updating the states, whereas, the RNDDR solves a nonlinear optimization problem to update the states.

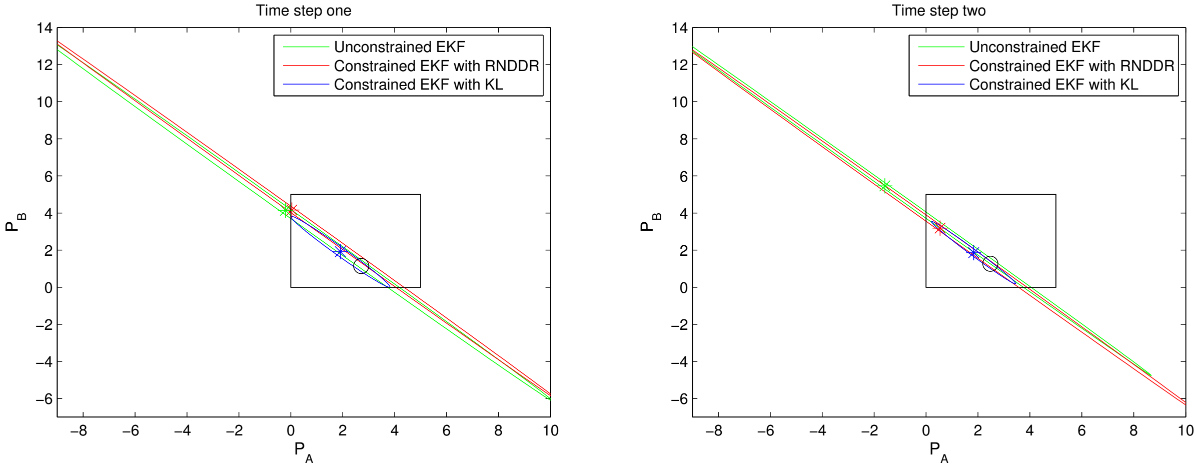


Figure 3.6: State space showing the state estimates and error covariance obtained using Unconstrained EKF, RNDDR, and KL methods (a) at first time step; (b) at second time step. True value of the states is marked by a black circle. The markers shown in the center of ellipse denote the respective state estimates obtained using the method

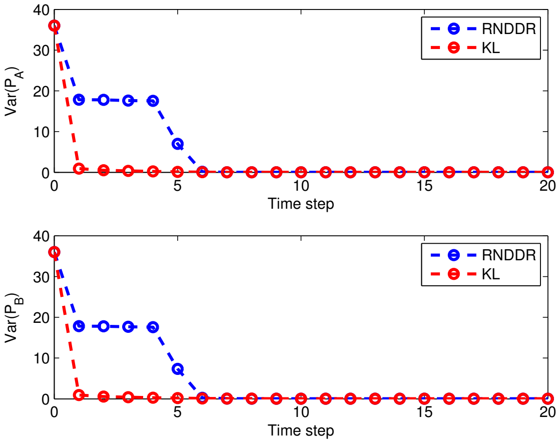


Figure 3.7: Evolution of error variances of P_A and P_B

Table 3.3: Performance comparison of RNDDR and KL methods

		RNDDR	KL
RMSE	P_A	0.7220	0.1417
	P_B	0.7422	0.1613
CPU(s)		15.5	197.3

Constrained EnKF:

The EnKF with the ensemble size of 100 particles is used to estimate the two states for 80 time steps. As we can see from Figure 3.8, same as the unconstrained EKF, the unconstrained estimation using the EnKF cannot converge to the real state trajectory owing to the large initial error. In the following, three methods are implemented to incorporate the inequality constraints above on the states.

The first method is the KL divergence based approach. The constrained mean and covariance are obtained using the algorithm describe in Table 3.2. As there is a strong correlation between P_A and P_B, a whole new ensemble is regenerated after the constrained mean and covariance are obtained using the KLD based approach. The second one is to apply the RNDDR optimization on the each of the individual particles in the posterior ensemble using Eq.(3.19). The third one is to apply the RNDDR optimization only on the estimated unconstrained mean using Eq.(3.20) and shift the whole posterior ensemble in parallel using Eq.(3.22). Figure 3.9 shows the constrained estimates at every step of these three approaches in comparison with the true state trajectory. Table 3.4 shows the root mean square error(RMSE) of the three methods as well as their computation time.

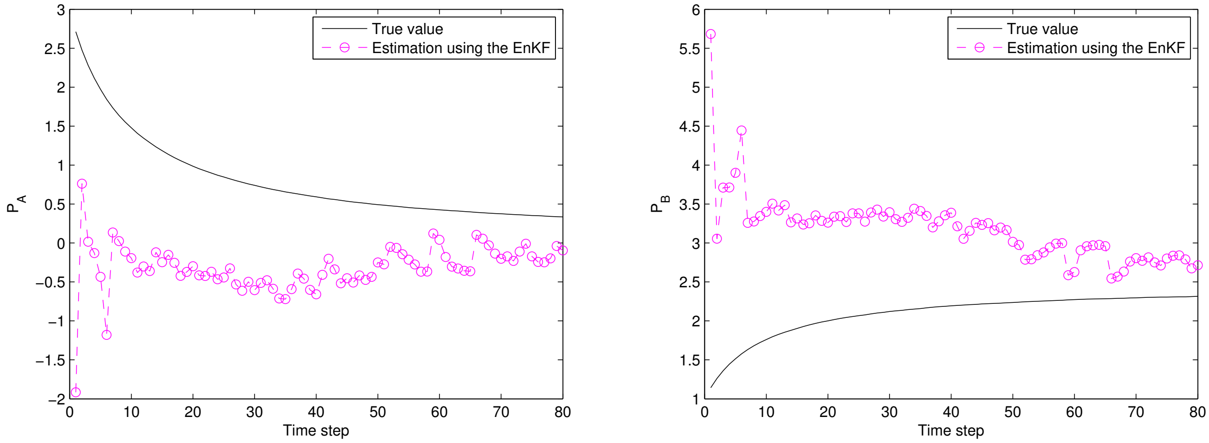


Figure 3.8: Estimation with unconstrained EnKF

Figure 3.9 shows that all three constrained approaches are able to provide converged estimation results. Together with Table 3.4, we can see that the RNDDR formulated on the

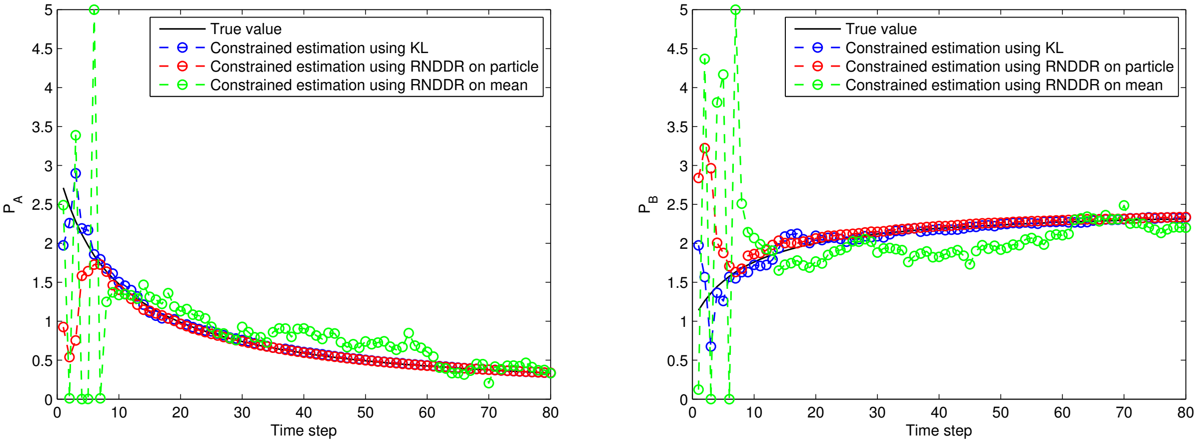


Figure 3.9: Constrained estimation using:KLD based approach(blue), RNDDR applied on individual particles(red) and RNDDR applied on mean(green).

Table 3.4: Comparison of three constraints implementation methods for EnKF and the KLD based constrained EKF.

Performance	Constrained EnKF		
	Proposed KL based method	RNDDR on unconstrained particles	RNDDR on unconstrained mean
RMSE(P_A)	0.1150	0.3486	0.6226
RMSE(P_B)	0.1373	0.3523	0.7356
CPU time(s)	283.48	583.55	7.68

mean exhibits the worst performance. For this method, figure 3.10 shows the two-dimensional distribution of particles before and after enforcing the constraints. The markers represent the point estimates, i.e. the mean value. The rectangle represents the two-dimension inequality constraints. As is explained before, for RNDDR applied on the mean, shifting the covariance in parallel does not guarantee the particles to fall within the constrained region. In figure 3.10, it is obvious that a large portion of the constrained ensemble remains outside the constraints, although the mean is pulled into the constraints. This means the constraining step does not help much to push the particles to the true value, which probably produces an estimate that is far from the true value, hence the large oscillations before convergence and the slow convergence rate.

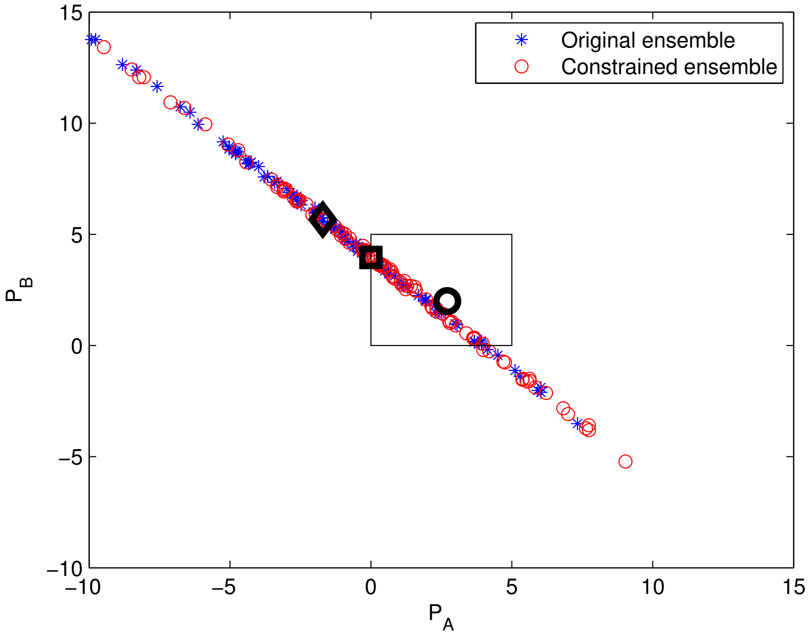


Figure 3.10: 2D distribution of particles using the RNDDR applied on the mean before and after the enforcing the constraints.(Diamond marker:unconstrained estimate; Square marker:constrained estimate; Circle marker:true value of states)

Figure 3.11 shows the two-dimensional distribution of the ensemble after enforcing constraints at step one using the RNDDR applied on the particle(red dots) and the KL based method(green dots). Note that this is the estimation result of time *step one*. The repre-

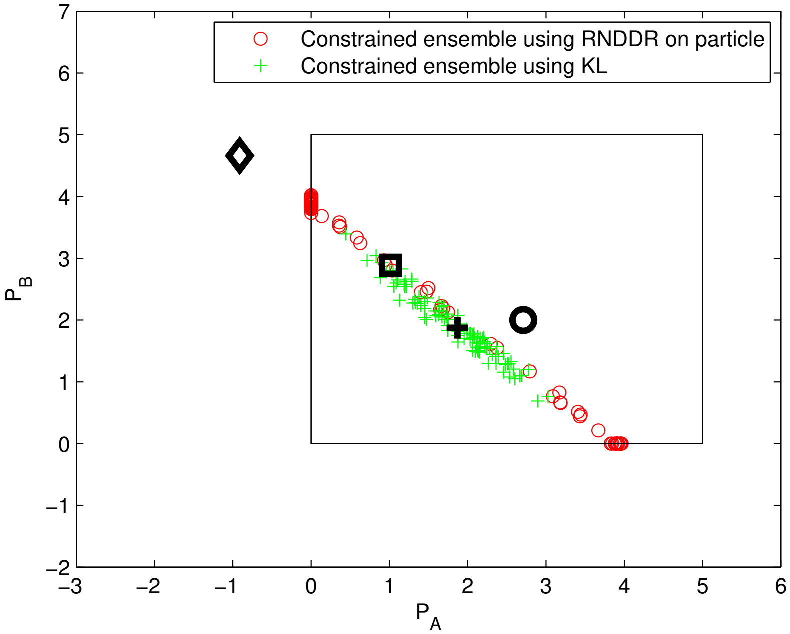


Figure 3.11: At time step one, 2D distribution of particles after enforcing constraints using the RNDDR applied on the particles and the KLD approach.(Diamond marker:unconstrained estimate; Square marker:constrained estimate using the RNDDR on the particle; Plus marker:constrained estimate using the KL; Circle marker:true value of states)

sentation of the markers is explained in the caption of the figure. Both methods force the particles in the original ensemble to go back to the constrained region, which enables both methods to produce a constrained estimate that is much closer to the true value than the previous RNDDR approach that is applied on the mean of the unconstrained estimate. The KL method evenly distributes the constrained ensemble on the ellipse whose mean and covariance are obtained from the constraining step. The RNDDR method, however, does not evenly distribute the particles because many of them sit close to the boundary of the constrained region. This is because the solution of the RNDDR is essentially a weighted summation of the error caused by deviation from the unconstrained particle $(\hat{x}_c - \hat{x})P^{-1}(x)(\hat{x}_c - \hat{x})^T$ and by the deviation from the measurement $(y_{obs} - h(\hat{x}))W^{-1}(y_{obs} - h(\hat{x}))^T$. If the unconstrained particles are too far away from the boundary, they tend to compensate the error caused by the first term by sitting on the boundary. Therefore the ensemble of produced by the KLD constraining method gets a stronger push to true value than the one from the

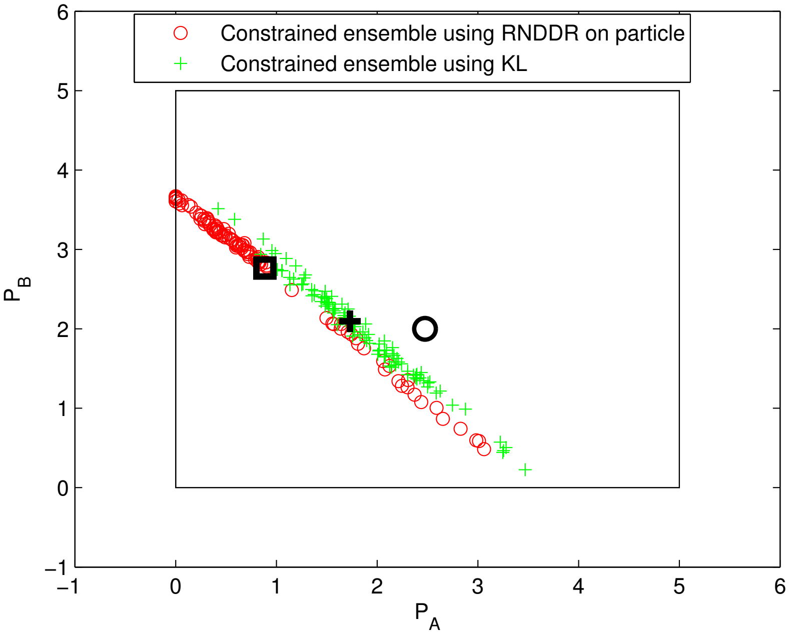


Figure 3.12: At time step two, 2D distribution of particles after enforcing constraints using the RNDDR applied on the particles and the KLD approach. (Square marker:constrained estimate using the RNDDR on the particle; Plus marker:constrained estimate using the KL; Circle marker:true value of states)

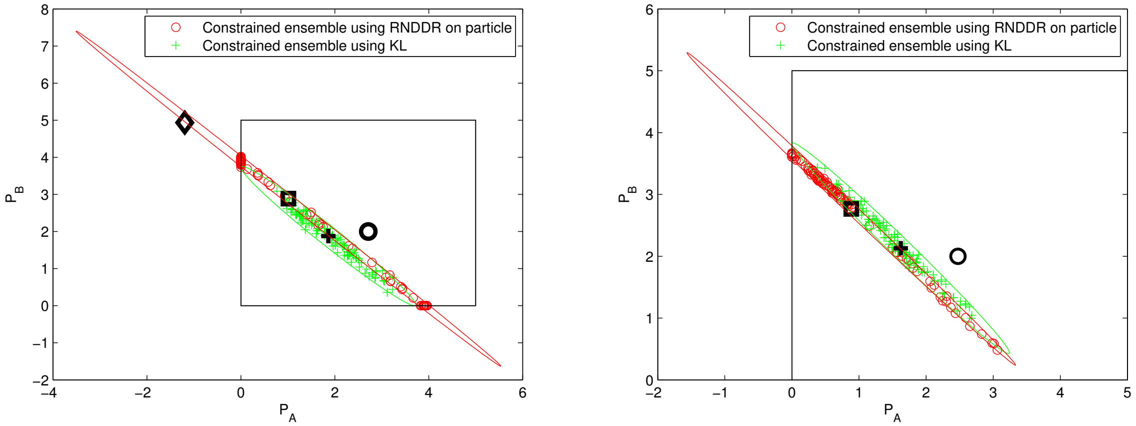


Figure 3.13: Contour of distribution of time step one and two after enforcing constraints using the RNDDR applied on the particles and the KLD approach. (Diamond marker:unconstrained estimate; Square marker:constrained estimate using the RNDDR on the particle; Plus marker:constrained estimate using the KL; Circle marker:true value of states)

RNDDR approach, especially at the first few steps when the initial ensemble is poor. This is the reason why the RNDDR method has relatively large oscillations at initial few steps compared to the KLD based method, hence larger estimation error. Figure 3.13 explains the better performance of the KL based approach from another angle. Figure 3.13 shows the contours of the constrained Gaussian distribution using the two aforementioned constrained methods at time step one and two. The Gaussian produced by the KLD method is completely within the constraints, while the one produced by the RNDDR is partially outside constraints. Therefore the KLD method converges to the true value faster.

Figure 3.12 shows the distribution of the constrained ensemble at *time step* two using the KLD method and RNDDR on the particle method. Compared to step one, most of the particles obtained by the RNDDR method(red dots) are pulled into the boundary, closer to the true value. However, compared to the particles obtained by the KLD method(green dots), they are still further away from the true value. The simulation results of time step two further illustrate why the RNDDR method generates larger oscillations than the KL method.

Another major disadvantage of the RNDDR applied on the particle method is the computation time. Table 3.4 also presents the comparison of total operation time of the three methods. We can see that the RNDDR applied on the particle consumes almost twice the time as the KLD based method. This is because the latter method solves only one optimization every time step, while the former has to solve 100 optimizations at every time step. The RNDDR applied on the mean method is significantly faster than the other two, yet gives worst estimation.

Remarks:

For this two-state batch reaction process, both the proposed constrained EKF and EnKF are applied. For the constrained EnKF, we compared three different methods for constraints implementation. We have the following conclusions based on the simulation results.

1. The KLD based constrained EKF obviously greatly outperforms the RNDDR based

EKF. This is because the KL method attempts to take both of the decision parameters of the Gaussian state distribution, namely mean and covariance, into consideration when dealing with the constraints. It formulates a convex optimization problem to penalize the deviation of whole distribution instead of just the mean in the RNDDR, with both mean and the covariance as the decision variables. Thus, the KL method is shown to have faster convergence rate.

2. The KLD based constrained EnKF outperforms the KLD based EKF in the sense of smaller RMSE. This is because of the superiority of the EnKF itself to the EKF to cope with nonlinearity.

3. The KLD based constrained EnKF outperforms the RNDDR based EnKF in either of the two formulations. This is because the KLD pushes the particles of the EnKF closer to the true value faster. Moreover, it also consumes less computation time.

3.5.2 Case study: Three-state continuous stirred tank reaction(CSTR)

We consider a CSTR gas reactor studied in [37],[94] and [54]. The reaction is given by:

$$\begin{aligned} A &\xleftrightarrow{k_1,k_2} B + C \\ 2B &\xleftrightarrow{k_3,k_4} C \end{aligned}$$

where $[k_1 \quad k_2 \quad k_3 \quad k_4] = [0.5 \quad 0.05 \quad 0.2 \quad 0.01]$, with a stoichiometric matrix:

$$v = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \end{bmatrix}$$

and a reaction rate:

$$r = \begin{bmatrix} k_1 C_A - k_2 C_B C_c \\ k_3 C_B^2 - k_4 C_c \end{bmatrix}$$

The first-principle model for the isothermal CSTR is:

$$\dot{x} = \frac{Q_f}{V_R} C_f - \frac{Q_0}{V_R} x + v^T r \tag{3.57}$$

where $Q_f = Q_0 = 1, V_R = 100$ and $C_f = [0.5 \quad 0.05 \quad 0]$.

The states of the process is $x = [C_A \quad C_B \quad C_c]$, which are the concentration of A,B and C. The measurement model is given by:

$$y = [RT \quad RT \quad RT]x \tag{3.58}$$

where R is the ideal gas constant and T is reactor temperature, RT=32.84.

The system is discretized with a sampling time of $\Delta t = 0.25$ and simulated for 100 time steps. The initial value of the states is $x_0 = [0.5 \quad 0.05 \quad 0]$ and the estimation of the three states starts with a poor initial guess $\hat{x}_0 = [0 \quad 0 \quad 3.5]$, with a covariance matrix $P_0 = diag(4^2, 4^2, 4^2)$. The process noise is $w \sim N([0 \ 0 \ 0], 10^{-6} I_3)$, where I_3 is the identity matrix, and measurement noise is $v \sim N(0, 0.25^2)$.

Figure 3.14 shows the true state trajectory of the three states over 100 steps and the estimation using the unconstrained EnKF. Obviously, the estimation result of the unconstrained EnKF does not converge to the true value because of the poor initial guess. In order to achieve estimation that can be converged to the true value, it is necessary to add constraints to the state space. The following inequality constraint is put on the state variables:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} C_A \\ C_B \\ C_C \end{bmatrix} \leq \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$$

Figure 3.15 shows the constrained estimation results using the KLD based approach, the RNDDR applied on the individual updated particle and RNDDR applied on the updated mean respectively. Table 3.5 lists the RMSE and the computation time of the three methods. Same as the previous case study, the RNDDR applied on the updated mean yields the poorest performance in the sense of large oscillations and slow convergence rate. In fact, C_B and C_C do not show a converged estimation within 100 time step. It does have the fastest computation because it only solves one optimization problem every time step. The RNDDR applied on the particle loses to the KLD based method on the RMSE mainly because it produces larger oscillations and slower convergence rate than the KLD method at the initial few steps. This case study proves again that:

1. The added constraints serve as extra knowledge to help pull the unconstrained particles which take values far from the true value closer to the true value. That's to say, the constraints compensate for the lack of prior information of the states;

2. When enforcing the constraints, the KLD method pushes the posterior particles to the true value faster than the other two, which explains its better performance.

Table 3.5: Comparison of three constraints implementation methods.

State	Proposed KL based method	RNDDR on posterior particles	RNDDR on posterior mean
RMSE(C_A)	0.0250	0.0295	0.1068
RMSE(C_B)	0.0131	0.0323	0.1969
RMSE(C_C)	0.0246	0.0503	0.1974
CPU time(s)	351.22	610.54	8.56

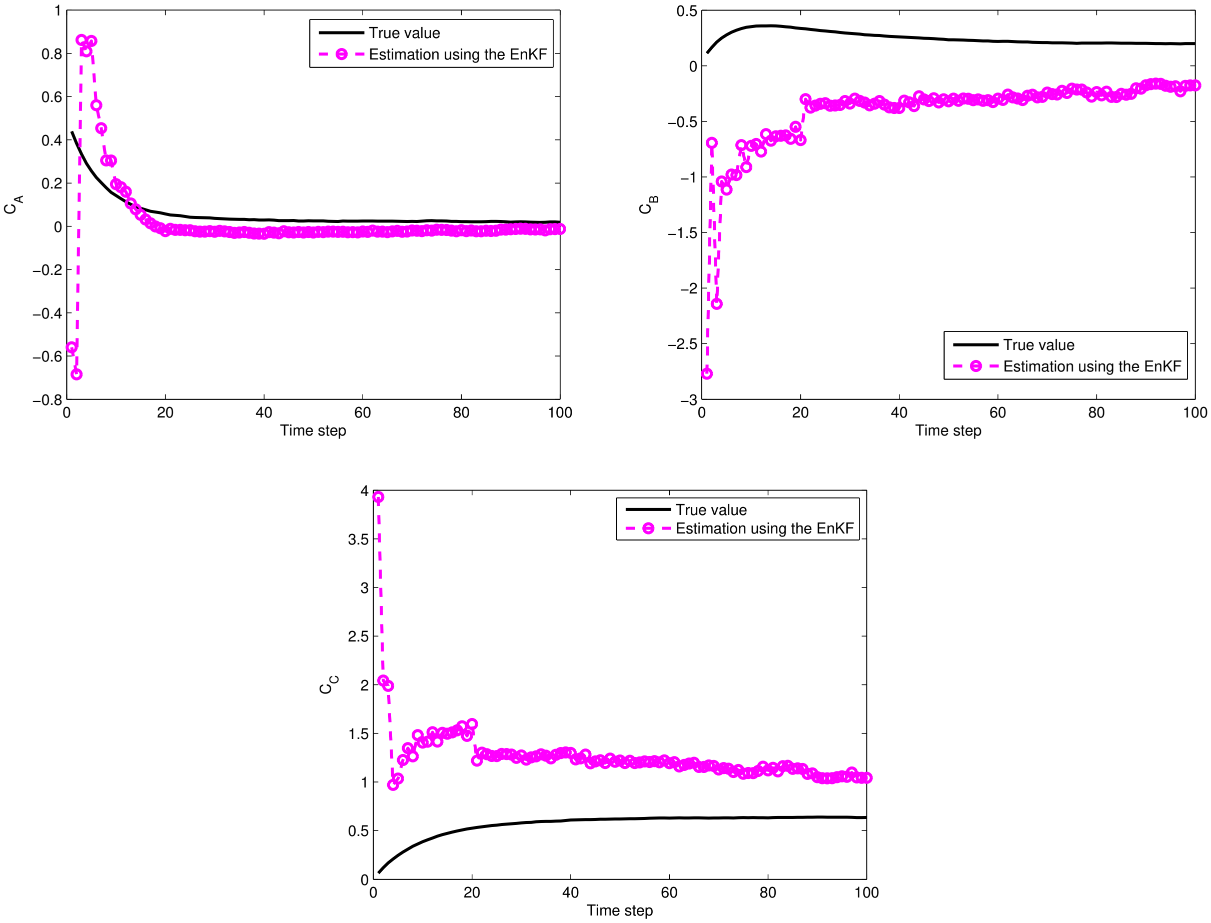


Figure 3.14: True state trajectory and estimation results of concentrations of A,B and C with unconstrained EnKF

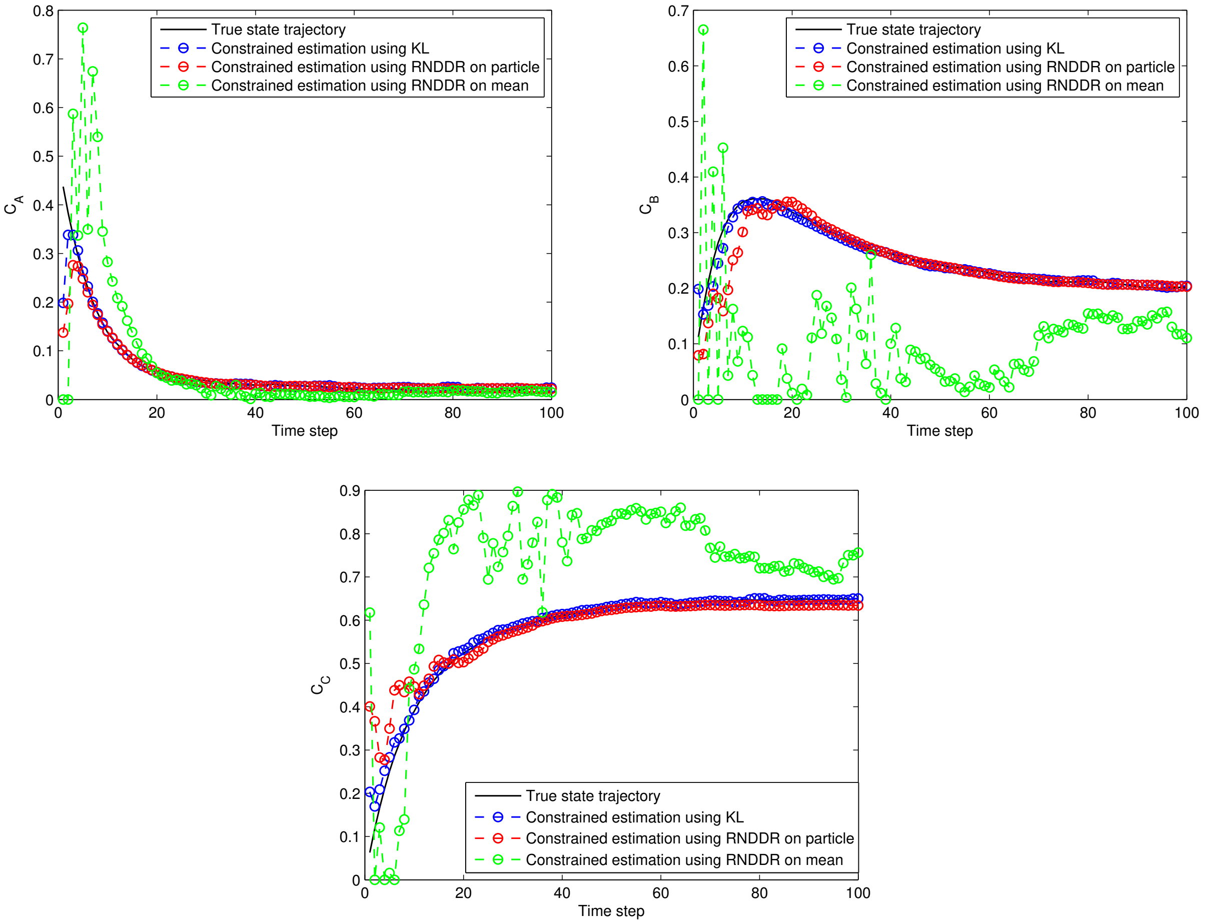


Figure 3.15: Constrained estimation results using the KLD based approach, the RNDDR applied on the particle and RNDDR applied on the mean

3.6 Conclusions

In this chapter, we proposed a constrained state estimation technique based on the KL divergence measure. In principle, this approach can be applicable to all Gaussian filters. Specifically, we formulated the constrained frameworks for two typical Gaussian filters, the EKF and EnKF. For both of these two filters, the proposed constrained approach is able to converge to the true values faster than the existing RNDDR approach . The single time step simulations demonstrate that the superiority of our approach is achieved due to constraining the error covariances inside the feasible region besides the mean value, which gives more accurate prediction of the state estimates. For each individual optimization, our approach indeed consumes more time since we solve conic optimization problem in contrast to the nonlinear optimization of the RNDDR. However, we optimize for the complete distribution in the KL formulation. On the other hand, the RNDDR either achieves partial information of the constrained distribution in a single run, or we have to perform the optimization multiple times to obtain the full distribution.

Chapter 4

Constrained multimodal ensemble Kalman filter based on Kullback-Leibler(KL) divergence

4.1 Introduction

Over the years several Bayesian nonlinear filters have been proposed to extend the well-known Kalman filter to nonlinear systems, such as the extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Ensemble Kalman Filter (EnKF). They all attempt to obtain a suboptimal solution by approximating the posterior distribution as Gaussian. However, due to the nonlinearity of the model and uncertainties such as process disturbances and measurement errors, the probability density function (PDF) of the states is multimodal, hence the Gaussian assumption of posterior PDF is not valid. Further, when the posterior PDF is strongly multimodal or heavily skewed, the Gaussian filters mentioned above will lead to inaccurate state estimates.

To deal with multimodality, particle Filter (PF) is the most widely used sample-based nonlinear Bayesian filter for non-Gaussian state estimation [30]. The PF can characterize any arbitrary posterior PDF of states based on the Monte Carlo (MC) samples associated with weights generated using the importance sampling (IS). The sequential importance sampling (SIS) builds a recursive framework for the PF to propagate the particles and update the weights sequentially. This Monte Carlo representation of the posterior PDF is ideal when the number of the particles is sufficiently large and the weights are assigned properly. However, the PF suffers from the degeneracy problem, which occurs when most of the particles have negligible weights except for one after a few iterations. Further, Shenoy et al. [84] also noted that the PF is more sensitive to model-plant mismatch than the Kalman based filters. The model-plant mismatch causes almost all particles to lie at the tail of the likelihood density, so that all weights collapse to zero after a certain time.

In recent years, approximating the state PDF using the Gaussian mixture model (GMM) has drawn increasing attention as it can provide a parametric model for the state PDF, compared to non-parametric representation using the sample based particle filters. In [1], a Gaussian sum filter (GSF) is proposed which approximates the state PDF using the Gaussian mixture model (GMM). The GSF performs multiple EKF in parallel to update the mean and covariance of each component of the GMM but the mixture component weights are unchanged before and after the update. Likewise, in [55] a Gaussian sum particle filter (GSPF) is proposed by using Gaussian mixture approximation of the PDF. In this case, it uses a bank of Gaussian particle filters, which removes the resampling step by assuming that the prior and posterior state PDF to be Gaussian.

As the ensemble Kalman filter (EnKF) is especially promising for high-dimensional systems compared to other filters, there have been several studies to extend the traditional EnKF to cope with multimodal systems. In [90], Smith proposed a clustering EnKF to handle the multimodal state PDF approximated by GMM. The main idea is to obtain a parametric GMM model at the prediction step using the expectation maximization (EM) algorithm, and then update the parameters of the GMM by assimilating the measurements. However, the posterior PDF after the update step is approximated as Gaussian distribution.

In our recent work [60], we proposed a modified algorithm such that the multimodality of the posterior distribution is maintained. In [19], the EnKF is also extended to multimodal systems. An EM clustering is also performed to obtain a GMM model at the prediction. Then it proposes to assign each particle of the prior and posterior ensemble to a certain mode based on the mode weights, such that the particles assigned to each mode can be updated using the EnKF. At the update step, the membership of each particle changes correspondingly with the update of the mode weights. Same as our work, the posterior PDF is retained to be the Gaussian mixture . These EnKF based GMM filters (henceforth called as GMM-EnKF) can provide a better approximation of the nonlinear dynamics of the systems over the EKF based GSF, and they do not have the collapsing problems as in the case of GSPF.

It should be noted that in conventional recursive state estimation algorithms, the constraints on process variables are often ignored. However, in practical problems, it is imperative to account for the constraints in state estimation procedure such that the estimate is physically meaningful, and satisfies operational requirements. There has been extensive work on constrained state estimation of Gaussian filters. In [99] a novel recursive nonlinear dynamic data reconciliation (RNDDR) technique is proposed to incorporate constraints into the recursive Bayesian filtering framework. The RNDDR approach essentially attempts to project the unconstrained estimate into the constrained region by solving a quadratic or nonlinear programming problem. Later, in [98] a variant of RNDDR for the Unscented Kalman filter is proposed, by suitably constraining the individual sigma points. On the other hand, in [71] a constrained EnKF is proposed by constraining individual particle using the RNDDR. To the best of our knowledge, there is limited work on the constrained state estimation for non-Gaussian systems. The most studied constrained non-Gaussian filter is the constrained particle filter. In [72] and [83], several optimization formulations are proposed to constrain individual particles of the PF using RNDDR approach.

In this work, we attempt to achieve the constrained estimate by putting the inequality constraints on the parameters of the GMM. Most of the previous work in the literature consider spacial constraints, especially in the study of image segmentation. They focus on improving the unsupervised clustering of GMM by considering spatial constraints among the neighboring pixels. In [10], a spatially variant finite mixture mode is proposed as a modification of the finite Gaussian mixture model to assign random values to pixel labels using the Gibbs function. In [32], each tissue of the brain is modeled with a mixture of many oriented Gaussian and incorporates the global constraint, the intensity of a tissue, into the modeling step. A modified expectation maximization algorithm(EM) is presented for this purpose. Although the work in [86] does not relate to segmentation, it considers the constraints called equivalence constraints which are incorporated specifically in the process of data clustering. Equivalence constraints are defined as positive constraints if the data pairs arise from the same cluster and negative if from the different clusters. There have been works which study inequality constraints for GMM. Most of them attempt to implement the constraints in the M step of the EM algorithm. However, direct optimization of the complete data likelihood in the M step with parameters subject to constraints is usually intractable. To this end, an expectation-conditional maximization algorithm(ECM) [67] was proposed that decomposes the M step into many simpler conditional maximization steps which are computationally tractable. However, computational cost of solving such optimization algorithms are usually high, and hence not desirable for online filtering problems.

The constrained state estimation procedure is based on the GMM-EnKF developed in chapter 2. The objective of this chapter is to incorporate constraints in this GMM-EnKF framework. For this purpose, we project the unconstrained distribution into the constrained region by minimizing the Kullback-Leibler divergence (KLD) between Gaussian mixture distributions. The profound challenge in the constrained GMM-EnKF is on how one reshapes the GMM according to the constraints. In other words, we develop an optimization algorithm to impose inequality constraints on the posterior distribution of the GMM-EnKF introduced previously, so that the targeted constrained posterior PDF stays within the constraints without losing the information gained from the unconstrained filtering step. The main contributions in this chapter are as follows: (a) since there is no closed form expression for KLD between two Gaussian mixture distributions, we present an approximation of the KLD that is suitable for solving the constrained filtering problem, (b) formulate the constrained GMM-EnKF algorithm using convex optimization tools, and (c) owing to the non-convexity of the objective function, we propose a two step optimization procedure, where at each step we solve a convex optimization problem that ensures global optimality.

The rest of this chapter is organized as follows: Section 4.2 reviews the GMM-EnKF filter proposed in chapter 2. Section 4.3 defines the problem statement of the constrained state estimation. Section 4.4 discusses the different distance measures between the unconstrained and constrained posterior distribution, and highlights the significance of KL divergence. In section 4.5, we propose a modification of Goldberger's approximation of KLD between two Gaussian mixture distributions, and formulate the constrained EnKF state estimation problem. Further, a two step optimization approach is presented. Section 4.6 presents the demonstration of the proposed methodology through two case studies.

4.2 Review of the multimodal ensemble Kalman filter (GMM-EnKF)

At each time step k, the GMM-EnKF has two steps-forecast and update. The forecast step is identical to the EnKF. An ensemble of size N, The forecast step is identical to the EnKF. An ensemble of size N, $\{x_k^i\}_{i=1,\cdots,N}$, is drawn from the prior distribution of the states and forwarded through the model to obtain a predicted ensemble for the next time step. Then, the EM algorithm is performed on the predicted ensemble to obtain the estimates of the GMM with N_c components. Next, the Kalman update is performed based on each component in the GMM to get an ensemble of size $N \times N_c$. Finally, these ensemble members are combined based on their weights and reduced to a size of N. The details of the algorithmic sequence are as follows:

Forecast:

1. The first portion of the forecast step is to determine the number of components N_c in the multimodal distribution. N_c can be determined using the Bayesian or other information criteria(BIC) [41], or using prior knowledge.

2. The prior ensemble $x^i_{ki=1,\cdots,N}$ is propagated through the model to get the predicted values of the ensemble $x^f_k, i_{i=1,\cdots,N}$.The EM algorithm is applied on $\{x^{f,i}_k\}_{i=1,\cdots,N}$ to give us the parameters of the prior distribution ($\tau^f_{k,j}, \mu^f_{k,j}andP^f_{k,j}$) of each component j.

$$p(x^f_k) = \sum_{j=1}^{N_c} \tau^f_{k,j} p_j(x^f_k) = \sum_{j=1}^{N_c} \tau^f_{k,j} N(x^f_k; \mu^f_{k,j}, P^f_{k,j}) \tag{4.1}$$

Update:

3. For each component j of the distribution, the Kalman gain matrix for each Gaussian component is computed by utilizing the membership probability matrix W.

$$P[j]^f H^T = \sum_{i=1}^{N} w_{i,j}(x^{f,i}_k - \mu_{k,j})(Hx^{f,i}_k - H\mu_{k,j})^T / n_j \tag{4.2}$$

$$HP[j]^f H^T = \sum_{i=1}^{N} w_{i,j}(Hx^{f,i}_k - H\mu_{k,j})(Hx^{f,i}_k - H\mu_{k,j})^T / n_j \tag{4.3}$$

$$K[j] = P[j]^f H^T (HP[j]^f H^T + R)^{-1} \tag{4.4}$$

where $w_{i,j} = \frac{\tau^f_{k,j} N(x^{f,i}_k; \mu^f_{k,j}, P^f_{k,j})}{\sum_{m=1}^{N_c} \tau^f_{k,m} N(x^{f,i}_k; \mu^f_{k,m}, P^f_{k,m})}$, $n_j = \sum_{i=1}^{N} w_{i,j}$ and H is the linearized measurement operator.

4. In the update step, assuming one Gaussian component j claims the ownership of all the ensemble members, the Kalman update can be performed for each component member under component j. This gives us an ensemble size of $N \times N_c$.

$$x^{a,i}_{k,j} = x^{f,i}_k + K[j](d - Hx^{f,i}_k - v_i) \tag{4.5}$$

where v_i is the measurement noise.

5. The $N \times N_c$ ensemble members can be combined to form N members by using the probability matrix W. W is the matrix of $w_{i,j}$. This gives us the final posterior ensemble $\{x_k^{a,i}\}_{i=1,\cdots,N}$.

$$x_k^{a,i} = \sum_{j=1}^{N_c} w_{i,j} x_{k,j}^{a,i} \tag{4.6}$$

6. The mean and covariance of the posterior distribution can be computed as:

$$\mu_{k,j}^a = \sum_{j=1}^{N} w_{i,j} x_{k,j}^{a,i} / n_j \tag{4.7}$$

$$P_{k,j}^a = \sum_{j=1}^{N} w_{i,j} (x_{k,j}^{a,i} - \mu_{k,j}^a)(x_{k,j}^{a,i} - \mu_{k,j}^a)^T / n_j \tag{4.8}$$

7. The posterior weight of each component of the distribution can be computed based on the observed data d, which contains the measurements y.

$$\tau_{k,j}^a = p(\mu_{k,j}^a, P_{k,j}^a, R|d) = \frac{p(d|\mu_{k,j}^a, P_{k,j}^a, R) n_j}{\sum_{m=1}^{N_c} p(d|\mu_{k,j}^a, P_{k,j}^a, R) n_j} \tag{4.9}$$

$$p(d|\mu_{k,j}^a, P_{k,j}^a, R) = \frac{exp[-\frac{1}{2}(d - H\mu_{k,j}^a)^T (HP_{k,j}^a H^T + R)^{-1} (d - H\mu_{k,j}^a)]}{\sqrt{(2\pi)^m |HP_{k,j}^a H^T + R|}} \tag{4.10}$$

8. With the estimates of the parameters of the GMM $\mu_{k,j}^a, P_{k,j}^a \text{and} \tau_{k,j}^a$, we have a full distribution of the posterior state distribution. The point estimate at time k according to the posterior PDF is given by:

$$x_k^a = \sum_{j=1}^{N_c} \tau_{k,j}^a \mu_{k,j}^a \tag{4.11}$$

4.3 Problem statement

Let us denote the states, inputs and measurements of the system as x_k, u_k, and y_k at k^{th} time step, respectively. Consider a stochastic state space model of the system, which is given by:

$$x_k = f(x_{k-1}, u_{k-1}) + v_k \tag{4.12}$$

$$y_k = h(x_k) + w_k \tag{4.13}$$

where $f(.)$ and $h(.)$ denote the discrete time process model and measurement model. And, v_k and w_k denote the process noise and measurement noise, characterized by Gaussian distribution with zero mean and covariance matrices Q, and R, respectively.

A general framework for non-linear state estimation can be presented in terms of the recursive Bayesian filtering algorithm, which consists of the prediction and update step, and they can be mathematically expressed as follows:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{4.14}$$

$$p(x_k|y_{1:k}) \propto p(x_k|y_{1:k-1})p(y_k|x_k) \tag{4.15}$$

where $p(x_k|x_{k-1})$ denotes the state transition probability obtained from the nonlinear process model defined in (4.12), $p(y_k|x_k)$ denotes the likelihood function defined using (4.13), $p(x_k|y_{1:k-1})$ denotes the prior distribution of the state variables obtained at the prediction step, and $p(x_k|y_{1:k})$ denotes the posterior distribution of the state variables obtained at the update step. In the conventional EnKF, the prior and posterior distribution of the ensemble members of state variables are assumed to be Gaussian. However, due to the nonlinear process model, the resulting state distributions in the prediction as well as at the update step are multimodal. Therefore, the assumption of Gaussian distribution of the ensemble members is not valid, and hence we propose to use mixture distribution to approximate the priors and posteriors. Further, the conventional EnKF does not account for inequality constraints in the state estimation procedure. Therefore, the focus of this work is to develop a constrained state estimation algorithm for multimodal systems in the EnKF framework.

The gist of our approach is to embrace the constraints in updating the posterior distribution of states. In other words, it is important to consider all the parameters of the distribution model under consideration while incorporating inequality constraints. In this work, we use the GMM as a parametric model for the approximation of both the predicted and posterior state PDF. Recall that the update equations of unconstrained GMM - EnKF, (4.7) - (4.11), represent the multimodal uncertain region around the state estimate ($\hat{x}_{k+1|k+1}$). This signifies the multivariate mixture Gaussian distribution with j components where the individual components of the distribution are parameterized in terms of mean vector $\hat{x}_{j,k+1|k+1}$, and covariance matrix $P_{j,k+1|k+1}$ and the corresponding mixture weight is denoted by τ_j. The main idea of our approach is to project the unconstrained solution into the constrained region such that both the state and its estimation error lie within the constrained space. The unconstrained posterior PDF, denoted as $r(x)$, can be obtained from the EnKF GMM algorithm presented in subsection 4.2, and is given by:

$$r(x) = p(x_k|y_{1:k}) = \sum_{j=1}^{N_c} \tau_{k,j}^{a} N(x; \mu_{k,j}^{a}, P_{k,j}^{a}) \tag{4.16}$$

As is discussed previously, in general, some portion or the complete distribution might lie outside the constrained region. Therefore, in this work, we consider re-designing $r(x)$ by imposing inequality constraints of the form $lb < x < ub$. The resulting constrained state PDF $q(x)$ can be denoted by:

$$q(x) = \sum_{j=1}^{N_c} \tau_{k,j}^{c} N(x; \mu_{k,j}^{c}, P_{k,j}^{c}) \tag{4.17}$$

As the GMM-EnKF is a Monte Carlo sampling based filter, simply surrendering the final point estimate x_k^a to constraints is not enough since a large portion of the particles may still lie outside the constrained region. This might happen even though the mean of all the modes are within the constraints. $r(x)$ in figure 4.1 presents an example of two-dimensional GM distribution with two modes. It has a large portion outside the rectangular constraints although the mean of both modes are still within constraints. Therefore, we consider achieving the $q(x)$ such that the entire distribution is within the constraints, with a user-defined confidence limit. $q(x)$ of figure 4.1 demonstrates the target distribution that we aim to achieve. In other words, the goal of the constraining step is to obtain the parameters $\tau^c_{k,j}$, $\mu^c_{k,j}$ and $P^c_{k,j}$ in $q(x)$ such that a distance measure between the unconstrained posterior distribution ($r(x)$) and the constrained posterior distribution is minimized. Mathematically, the state estimation problem for multimodal systems can be expressed as:

$$\min_{\tau^c_{k,j},\mu^c_{k,j},P^c_{k,j}} D(q||r) \tag{4.18}$$

$$\text{s.t.} \quad \mu^c_{k,j}, P^c_{k,j} \in \Theta; \quad j = 1, \ldots, N_c \tag{4.19}$$

$$0 \leq \tau^c_{k,j} \leq 1; \quad j = 1, \ldots, N_c \tag{4.20}$$

$$\sum_{j=1}^{N_c} \tau^c_{k,j} = 1 \tag{4.21}$$

where $D(q||r)$ denotes the distance measure between two distributions $q(x)$ and $r(x)$, and Θ signifies the constraint set defined by the inequality constraints.

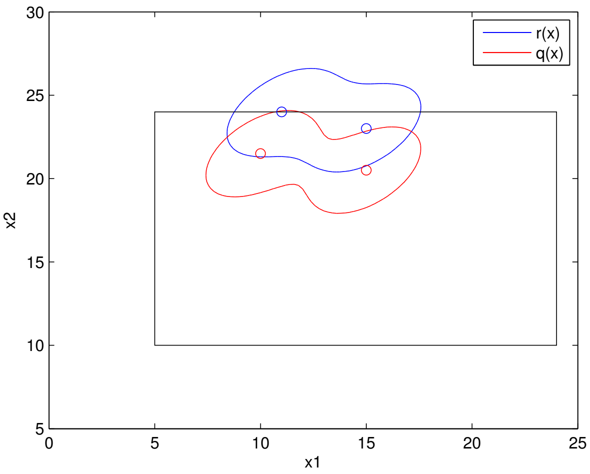


Figure 4.1: Unconstrained and constrained distribution with pre-specified confidence level and rectangular constraint region

4.4 Statistical metrics between probability density functions

As the filtering incorporates the information of the process, observation and prior knowledge into the estimation results, it is inappropriate to simply change the estimated GMM in order to satisfy constraints. We attempt to achieve a constrained distribution at the cost of the minimum information loss. Therefore it is necessary to select a cost function to measure the change of the overall distribution. In statistics, the probability distance metric is used to measure the similarity of two probability distributions. Various probability distance metrics have been developed for statistic distance calculation in different applications [13]. In the following, we first introduce several most widely used metrics to calculate the distance between two distribution $r(x)$ and $q(x)$ and then explain why we choose the KL divergence for our purpose.

1. **Bhattacharyya distance**

   The Bhattacharyya distance is given by:

$$D_B = -\ln \rho \tag{4.22}$$

where the the Bhattacharyya coefficient is given by:

$$\rho = \int \sqrt{r(x)q(x)dx} \tag{4.23}$$

The Bhattacharyya distance is often applied in signal processing in [46] to decrease the transmitting error by maximizing the difference of the distribution of a signal '0' and a signal '1'.

2. **L_p distance**

L_p distance can be defined as:

$$D_{L_P}(q, r) = (\int |r(x) - q(x)|^p dx)^{1/p} \tag{4.24}$$

where p normally takes the value 1 or 2.

3. **Kolmogorov distance**

Kolmogorov distance arises from the Kolmogorov-Smirnov test which determines whether two data sets differ significantly. Its definition is given by:

$$D_k = \int |r(x) - q(x)| dx \tag{4.25}$$

4. **Integral square difference**

Integral square difference (ISD) can be expressed as:

$$D_s = \int (r(x) - q(x))^2 dx \tag{4.26}$$

Compared to the Kolmogorov distance, the ISD puts more weights on the points where $r(x)$ and $q(x)$ have larger difference than the ones with smaller difference. Therefore the minimization of the ISD prioritizes on minimizing the largest difference created by the approximation distribution [103].

5. **KL divergence**

The KL divergence (KLD) is known as the relative entropy between two distributions. The forward KLD definition is given by:

$$D_{KL} = \int r(x) \log \frac{r(x)}{q(x)} dx \tag{4.27}$$

The most common application of the probability distance metric is to find an approximation distribution $q(x)$ for the original distribution $r(x)$, which is also the purpose in our work. Although all of the five metrics have found successful applications to achieve this goal in different areas, we select KL divergence to be our cost function in the constraining problem for the reason that the other four metrics calculate the similarity based on the degree of overlapping of the two distributions. In other words, they cannot serve as a metric if the two distributions do not overlap at all. When the two distributions do not overlap, the Bhattacharyya distance is equal to the constant '0' simply because either $r(x) = 0$ or $q(x) = 0$. The Lp distance and the Kolmogorov distance are equal to the constant '1' because they will be simplified as $D_{LP} = D_k = \int |r(x)|dx = 1$ or $D_{LP} = D_k = \int |q(x)|dx = 1$. The ISD has the same problem. If the two distributions do not overlap, D_s is simplified to $\int r(x)^2 dx$ or $\int q(x)^2 dx$. If we recall that the product of two Gaussian distributions is still Gaussian, $r(x)^2$ and $q(x)^2$ are still Gaussian distribution, i.e. D_s will reduce to 1 if no common support exists between the two distributions. However, KL divergence can still reflect the difference between $r(x)$ and $q(x)$ when they do not overlap. The further they lie apart, the larger KL divergence will be. The overlapping issue is important to the constraining step because the constrained region could have very little or even no overlap with the unconstrained estimated distribution. The KL divergence can still serve well as the metric of probability distance in this case while the other four cannot.

The KL divergence itself is not a strict distance measure for the most prominent reason that it is not symmetric, i.e. $D_{KL}(r(x), q(x)) \neq D_{KL}(q(x), r(x))$. By extending KL divergence to $D_{KL}(r(x), q(x)) + D_{KL}(q(x), r(x))$, it satisfies the symmetric requirement. However, the asymmetric form is still widely used especially when the symmetric form has a complicated analytical form. As discussed in the previous chapter, the KL divergence has an analytical form if both $r(x)$ and $q(x)$ are Gaussian. Unfortunately, there is no availability of closed form expression for KL divergence between two Gaussian mixture distributions. Therefore, the focus of next subsection is to deal with the approximations of KL divergence between two mixture Gaussian distributions.

4.5 Design of the constrained GMM-EnKF

4.5.1 Approximations of KL divergence between Gaussian mixture distributions

In this subsection, we briefly discuss the different approximations for KL divergence between two Gaussian mixture distributions that are available in literature, and present the approximation that is suitable for our constrained state estimation problem.

Monte Carlo approximation

The Monte Carlo simulation is the most straightforward solution for calculating KL divergence between two mixture distributions. Rewriting the definition of KL in terms of expectations as:

$$\begin{aligned} D_{KL} &= \int r(x) \log \frac{r(x)}{q(x)} dx \\ &= \int r(x) \log r(x) - \int r(x) \log q(x) \\ &= E_{r(x)}[\log r] - E_{q(x)}[\log q] \end{aligned} \tag{4.28}$$

Using random samples $\{x_i\}_{i=1,\dots,N}$ from f(x), D_{KL} based on Monte Carlo approach can be calculated by:

$$D_{KL} = \frac{1}{N} \sum_{i=1}^{N} \log \frac{r(x_i)}{q(x_i)} \tag{4.29}$$

However, the Monte Carlo method causes heavy computation complexity, and further it is not suitable for defining the objective function in the constrained state estimation problem as it can calculate the measure when two mixture densities are given.

In [38], several other approximation approaches have been described. An unscented transformation approach uses sigma points to approximate the expectation term $E_{r(x)}[\log r]$. A Gaussian approximation method simply replaces $r(x)$ and $q(x)$ with Gaussian distributions and reduces the KLD between two Gaussian mixtures to two Gaussians, which is proved to be a crude and inaccurate method. Another matching modes based approximation first establishes a mapping mechanism between components in $r(x)$ and $q(x)$ and calculate the KLD of the two Gaussian mixtures based on the mapping pairs. Another category uses the Jensen's inequality to achieve a variational upper or lower bound of the original KLD.

Two criteria should be taken into consideration when searching for an appropriate approximation to the KLD in our constraining problem. The first is the accuracy of the approximation. As is pointed out in [38], out of the approximation approaches introduced above, the Monte Carlo approximation is shown to be the most accurate, followed by the unscented transform method. The variational method and matching modes based method have similar performance, with the variational method slightly better. The second factor is the complexity for the use in optimization as a cost function, which solves for the parameters of the approximated GMM, weights, means and covariance, under constraints. The complexity and computation burden of the optimization problem is especially important when it is incorporated into an online filtering framework. The Monte Carlo and unscented transform approximation are not appropriate to be used as cost function because it is difficult to solve such an optimization problem. In this work, we use an analytic upper bound provided by the Goldberger's matching modes based approximation as the objective function. In the following, we show that a convex optimization problem can be formulated based on this upper bound.

Goldberger's matching based approximation

Since there exists a closed form solution for the KLD between two Gaussians, the basic idea in approximating the KL divergence between two mixture of Gaussians is by finding the matching between the Gaussian components of the two mixture of Gaussian distributions. Assume $r(x)$ and $q(x)$ are both Gaussian mixture distributions and let $r(x) = \sum_{j=1}^{N_c} \tau_j^a r_j(x)$ and $q(x) = \sum_{l=1}^{N_m} \tau_l^c q_l(x)$. The reverse KLD of these two Gaussian mixtures can be written as:

$$
\begin{aligned}
D_{KL}(q||r) &= \int q(x) \log \frac{q(x)}{r(x)} dx && (4.30)\\
&= \int \sum_{l=1}^{N_m} \tau_l^c q_l(x) \log \sum_{l=1}^{N_m} \tau_l^c q_l(x) dx - \int \sum_{l=1}^{N_m} \tau_l^c q_l(x) \log \sum_{j=1}^{N_c} \tau_j^a r_j(x) dx && (4.31)\\
&= \sum_{l=1}^{N_m} \tau_l^c (\int q_l(x) \log \tau_l^c q_l(x) dx - \int q_l(x) \log \sum_{j=1}^{N_c} \tau_j^a r_j(x) dx) && (4.32)
\end{aligned}
$$

Assuming one term r_j in the sum $\sum_j \tau_j^a r_j$ which is proximal to q_l dominates the integral $\int q_l \ log \ r$, the KLD of the mixture can be approximated as follows:

$$
\begin{aligned}
&\approx \sum_{l=1}^{N_m} \tau_l^c (\int q_l(x) \log \tau_l^c q_l(x) dx - \max_j \int q_l(x) \log \tau_j^a r_j(x) dx) && (4.33)\\
&= \sum_{l=1}^{N_m} \tau_l^c \min_j (KL(q_l||r_j) + \log \frac{\tau_l^c}{\tau_j^a}) && (4.34)
\end{aligned}
$$

It should be noted that in order to obtain the approximation, we need to find a j that establishes a matching mechanism, denoted as $\pi(.)$, to match each mode in $q(x)$ with each mode in $r(x)$.

$$q_l(x) \rightarrow r_{\pi(l)}(x) \qquad (4.35)$$

where $r_{\pi(l)}$ represents the mode in $r(x)$ that $q_l(x)$ will be matched [28]. The matching between the individual modes in $q(x)$ and those in $r(x)$, i.e. $\{l = 1, \ldots, N_m\} \rightarrow \{j = 1, \cdots, N_c\}$ is achived by solving the following optimization problem:

$$
\begin{aligned}
\pi(l) &= \min_{\pi(l)} ((D_{KL}(q_l||r_{\pi(l)}) + \log \frac{\tau_j^c}{\tau_{\pi(l)}^a}))\\
&= \min_{\pi(l)} (D_{KL}(q_l||g_{\pi(l)}) - \log \tau_{\pi(l)}^c) && (4.36)
\end{aligned}
$$

It is important to note that in the above approximation both $q(x)$ and $r(x)$ are known distributions and the aim is to compute the KL divergence. However, we can further simplify the expression by assuming that the number of mixture components in the unconstrained distribution is equal to the number of components in the constrained distribution in our state estimation problem. Furthermore, in our state estimation problem, $q(x)$ is unknown, therefore, it is reasonable to assume that the j^{th} term of the sum $\sum_l \tau_l^c q_l$ which is proximal to r_j dominates the integral $\int r_j \; log \; q$. In other words, the j^{th} component of $r(x)$ is a matching mode of the j^{th} component of $q(x)$, $r_j \rightarrow q_j$. As a result, the determination of the matching function is not required for our purpose. Therefore, the approximation of KL for two mixture Gaussian density can be expressed as:

$$D_{KL}(q||r) = \sum_{j=1}^{N_c} \tau_j^c (D_{KL}(q_j||r_j) + \log \frac{\tau_j^c}{\tau_j^a}) \tag{4.37}$$

The advantage of this approximation is that it is expressed in terms of KL divergence between Gaussian distributions for which the analytical expression is readily available.

4.5.2 Proposed KL divergence based approach

In this subsection, we present the KLD based approach to handle inequality constraints in the state estimation procedure in the EnKF framework. The motivation of our proposed method is to embrace the constraints in updating both the point estimate(mean) and estimation error covariance matrix. The main idea of our approach is to project the individual components of the unconstrained Gaussian mixture solution onto the constrained region such that both the states and estimation errors lie within the constrained space. Since Kullback-Leibler (KL) divergence is known to be a measure of similarity between two distribution functions, we seek to determine the multivariate Gaussian distribution that is the projection of a particular component of the unconstrained mixture distribution in the constrained region that is close and similar to the unconstrained distribution obtained from the EnKF approach.

Let us denote the unconstrained and constrained posterior distribution of the GMM-EnKF filter at the k^{th} time step as $r(x) = \sum_{j=1}^{N_c} \tau_{k,j}^a r_j(x) = \sum_{j=1}^{N_c} \tau_{k,j}^a N(x; \mu_{k,j}^a, P_{k,j}^a)$ and

$q(x) = \sum_{j=1}^{N_c} \tau^c_{k,j} q_j(x) = \sum_{i=1}^{N_c} \tau^c_{k,j} N(x; \mu^c_{k,j}, P^c_{k,j})$, respectively. If $q_j(x)$ and $r_j(x)$ are the j^{th} components of the mixture Gaussian distribution, then the KL divergence of the individual components has a closed form expression which is given by:

$$\begin{aligned} D_K L(q_j || r_j) = \frac{1}{2}[log\frac{|P^a_{k,j}|}{|P^c_{k,j}|} + Tr({P^a_{k,j}}^{-1} P^c_{k,j}) - d \\ + (\mu^c_{k,j} - \mu^a_{k,j})^T {P^a_{k,j}}^{-1} (\mu^c_{k,j} - \mu^a_{k,j})] \end{aligned} \tag{4.38}$$

where the symbol $|M|$ denotes the determinant of the covariance matrix M, and d is the dimension of state vector.

Now, let us consider constraining the individual components of the mixture distribution to handle inequality constraints. Recall that $\mu^a_{k,j}$ and $P^a_{k,j}$ are the mean and error covariance of the j^{th} component of the unconstrained GMM-EnKF algorithm (4.7) -(4.8). It is important to note that the state covariance $P^a_{k,j}$ obtained from the GMM-EnKF algorithm signifies the ellipsoid representation around the mean $\mu^a_{k,j}$. Therefore, to constrain the state covariance, we utilize the following expression of ellipsoid:

$$\mathcal{E} = \{\mu^c_{k,j} + \alpha S_{k,j} z \mid \|z\|_2 \leq 1\} \tag{4.39}$$

where $S_{k,j}$ is the positive square root of $P^c_{k,j}$ and α depends on the confidence limit and it is prescribed by the user (e.g., $\alpha = 2$ signifies a confidence limit of 95%). In order to bound the state covariances, we enforce the following constraints:

$$\mathcal{E} = \{(x_{lb} \leq \mu^c_{k,j} + \alpha S_{k,j} z \leq x_{ub}) \mid \|z\|_2 \leq 1\} \tag{4.40}$$

or equivalently, the above constraint can rewritten as:

$$\tilde{\mu}_j := \mu^c_{k,j} + \alpha S_{k,j} z \mid \|z\|_2 \leq 1 \tag{4.41}$$

$$h_s^T \tilde{\mu}_j + t_s \leq 0; s = 1, \ldots, m \tag{4.42}$$

where h_s is the s^{th} row of the matrix $H = [I; -I]$ and t_s is the s^{th} element of vector $t = [x_{ub}; -x_{lb}]$. Now, the optimization formulation to determine the j^{th} component of state updates and state covariances of the constrained filtering problem can be formulated as follows:

$$\begin{aligned}
&\min_{\mu^c_{k,j}, P^c_{k,j}} D_{KL}(q_{k,j}||r_{k,j}) && (4.43)\\
&\text{s.t.} \quad S_{k,j} = P^{c1/2}_{k,j} && (4.44)\\
&\qquad \tilde{\mu}_j := \mu^c_{k,j} + \alpha S_{k,j} z \mid \|z\|_2 \leq 1 && (4.45)\\
&\qquad h_s^T \tilde{\mu}_j + t_s \leq 0; s = 1, \ldots, m && (4.46)
\end{aligned}$$

Now, the constrained state estimation problem based on the KL of the mixture can be formulated as:

$$\begin{aligned}
&\min_{\tau^c_{k,j}, \mu^c_{k,j}, P^c_{k,j}} D_{KL}(q||r) = \sum_{j=1}^{N_c} \tau^c_{k,j} \Big(D_{KL}(q_{k,j}||r_{k,j}) + \log \frac{\tau^c_{k,j}}{\tau^a_{k,j}} \Big) && (4.47)\\
&\text{s.t.} \quad S_{k,j} = P^{c1/2}_{k,j}; \quad j = 1, \ldots, N_c && (4.48)\\
&\qquad \tilde{\mu}_j := \mu^c_{k,j} + \alpha S_{k,j} z \mid \|z\|_2 \leq 1; \quad j = 1, \ldots, N_c && (4.49)\\
&\qquad h_s^T \tilde{\mu}_j + t_s \leq 0; s = 1, \ldots, m; \quad j = 1, \ldots, N_c && (4.50)\\
&\qquad 0 \leq \tau^c_{k,j} \leq 1; \quad j = 1, \ldots, N_c && (4.51)\\
&\qquad \sum_{j=1}^{N_c} \tau^c_{k,j} = 1 && (4.52)
\end{aligned}$$

The above formulation is a semi-infinite optimization problem, and it is not computationally tractable owing to the nonlinear matrix constraint (4.44) and infinite dimensional constraints (4.45).

4.5.3 Problem reformulation

In this subsection, we present the convex optimization techniques to reformulate the above infinite dimensional optimization problem such that it can be cast as a convex optimization problem. To this end, first let us introduce some of the definitions that will enable us to reformulate the problem. For more details on the definitions, the reader is referred to [12].

Definition 4.1: The epigraph of a function $f : R^n \rightarrow R$ is defined as [12]:

$$\textbf{epi f} = \{(x, t) | x \in \textbf{dom} f, f(x) \leq t\}$$

The epigraph of a function is the region above the function. For a convex function f, the minimization of f is equivalent to finding the lowest point of the epigraph.

Definition 4.2: For any symmetric matrix M of the form:

$$M = \begin{bmatrix} U & V \\ V^T & W \end{bmatrix}$$

The Schur complement of W is defined as $U - VW^{-1}V^T$. If W is invertible, then we have:

If $W \succ 0$, then $M \succeq 0$ if $U - VW^{-1}V^T \succeq 0$.

Theorem 4.1: The set of linear constraints $h_i^T z \leq t_i, z \in \mathcal{E}$ can be expressed as a set of second order cone constraints of the form $h_i{}^T z_{ss} + \alpha \|Sh_i\|_2 \leq t_i$.

Proof[12]: Recall that $z := z_{ss} + \alpha S\tilde{z}, ||\tilde{z}|| \leq 1$. Consider the infinite dimensional constraint:

$$\begin{aligned} & h_i^T z \leq t_i \; \forall z \in \mathcal{E} \\ \iff & \sup\{h_i^T z \mid z \in \mathcal{E}\} \leq t_i \\ \iff & \sup\{h_i^T (z_{ss} + \alpha S\tilde{z}) \mid \|z\|_2 \leq 1\} \leq t_i \\ \iff & h_i^T z_{ss} + \sup\{h_i^T \alpha S z \mid \|z\|_2 \leq 1\} \leq t_i \\ \iff & h_i^T z_{ss} + \alpha \|Sh_i\|_2 \leq t_i \end{aligned}$$

□

Theorem 4.2: The optimization problem (4.43) - (4.46) can be reformulated as the following conic optimization problem:

$$\min_{\mu^c_{k,j}, S_{k,j}, \gamma_j, Y_{k,j}} \frac{1}{2}[log\mathbf{det}(P^a_{k,j}) - d - 2log\mathbf{det}(S_{k,j}) + \gamma_j$$

$$+ (\mu^c_{k,j} - \mu^a_{k,j})^T {P^a_{k,j}}^{-1} (\mu^c_{k,j} - \mu^a_{k,j})] \tag{4.53}$$

$$\text{s.t.} \quad Tr(Y_{k,j}) \leq \gamma_j \tag{4.54}$$

$$\begin{bmatrix} Y_{k,j} & S_{k,j} \\ {S_{k,j}}^T & P^a_{k,j} \end{bmatrix} \succeq 0 \tag{4.55}$$

$$\alpha||S_{k,j}h_s|| + h_s^T \mu^c_{k,j} \leq t_s; \quad s = 1, \ldots, m \tag{4.56}$$

Proof:

Recall the definition of KLD between two Gaussian distributions from (4.38):

$$D_{KL}(q_j||r_j) = \frac{1}{2}[log\mathbf{det}(P^a_{k,j}) - d - log\mathbf{det}(P^c_{k,j}) + Tr({P^a_{k,j}}^{-1} P^c_{k,j})$$

$$+ (\mu^c_{k,j} - \mu^a_{k,j})^T {P^a_{k,j}}^{-1} (\mu^c_{k,j} - \mu^a_{k,j})] \tag{4.57}$$

It is important to note that $\mu^a_{k,j}$ and $P^a_{k,j}$ are the solution obtained from the unconstrained EnKF algorithm. Therefore, the first two terms are fixed values while solving the optimization problem, and the last term is convex with respect to the decision variable, $\mu^c_{k,j}$. Though the term $-log\mathbf{det}(P^c_{k,j})$ is convex, we let ${P^c}_{k,j} = S_{k,j}S^T_{k,j}$, to make the overall optimization problem convex. To this end, we use the properties of determinant $(det(AB) = det(A)det(B); det(A) = det(A^T))$, and the third term can be rewritten as $-2log\mathbf{det}(S_{k,j})$.

Now, consider the trace term, $Tr({P^a_{k,j}}^{-1} P^c_{k,j})$. Using the property that trace operator is invariant under cyclic permutations (i.e., $Tr(UVW) = Tr(VWU) = Tr(WUV) \neq$

$Tr(VUW)$), we can rewrite the trace term as:

$$Tr({P^a_{k,j}}^{-1} P^c_{k,j}) = Tr(S^T_{k,j} {P^a_{k,j}}^{-1} S_{k,j}) \tag{4.58}$$

Introducing auxiliary variables $Y_{k,j}$ and γ_j, we can use the definition of epigraph to express the trace term as:

$$\min_{Y_{k,j},\gamma_j} \gamma_j \tag{4.59}$$

$$s.t. \quad Tr(Y_{k,j}) - \gamma_j \leq 0$$

$$Y_{k,j} - S^T_{k,j} {P^a_{k,j}}^{-1} S_{k,j} \succeq 0 \tag{4.60}$$

Note that from Eq.(4.60) should be $Y = S^T_{k+1} P^{-1}_{k+1|k+1} S_{k+1}$, the matrix equality is strict with nonlinear term and hence difficult to optimize. Using the definition of Schur complement, the matrix inequality in Eq.(4.60) can be expressed as linear matrix inequality (LMI) as:

$$\begin{bmatrix} Y_{k,j} & S_{k,j} \\ S^T_{k,j} & P^a_{k,j} \end{bmatrix} \succeq 0 \tag{4.61}$$

These convex optimization tricks enable one to replace the trace term with a linear term, γ_j, in the objective function along with an upper bound for the trace term, and an LMI constraint.

Applying Theorem 4.1, the infinite dimensional constraints, (4.45) - (4.46), can be rewritten in terms of the following second order cone constraints [12]:

$$\alpha||S_{k,j} h_s|| + h^T_s \mu^c_{k,j} \leq t_s; \quad s = 1, \ldots, m \tag{4.62}$$

Now, the resulting optimization problem can be expressed using (4.53) - (4.56). □

From Theorem 4.2, the constrained state estimation problem based on the KLD of the mixture Gaussian can be recast as:

$$
\begin{aligned}
\min_{\tau^c_{k,j},\mu^c_{k,j},S_{k,j},\gamma_j} D_{KL}(q||r) = \sum_{j=1}^{N_c} \tau^c_{k,j}\Big(\frac{1}{2}[log\mathbf{det}(P^a_{k,j}) - d - 2log\mathbf{det}(S_{k,j}) + \gamma_j \\
+ (\mu^c_{k,j} - \mu^a_{k,j})^T {P^a_{k,j}}^{-1}(\mu^c_{k,j} - \mu^a_{k,j})] + \log\frac{\tau^c_{k,j}}{\tau^a_{k,j}}\Big) \qquad (4.63)
\end{aligned}
$$

$$\text{s.t.} \quad Tr(Y_{k,j}) \leq \gamma_j; \quad j = 1,\ldots,N_c \qquad (4.64)$$

$$\begin{bmatrix} Y_{k,j} & S_{k,j} \\ {S_{k,j}}^T & P^a_{k,j} \end{bmatrix} \succeq 0; \quad j = 1,\ldots,N_c \qquad (4.65)$$

$$\alpha||S_{k,j}h_s||+h_s^T\mu^c_{k,j} \leq t_s; \quad s = 1,\ldots,m; \quad j = 1,\ldots,N_c \qquad (4.66)$$

$$0 \leq \tau^c_{k,j} \leq 1; \quad j = 1,\ldots,N_c \qquad (4.67)$$

$$\sum_{j=1}^{N_c} \tau^c_{k,j} = 1 \qquad (4.68)$$

All the constraints are convex in terms of decision variables in the proposed optimization problem. However, the objective function is non-convex due to the product of $\tau^c_{k,j}$ with the terms in $D_{KL}(q_j||r_j)$. As a consequence, the overall optimization problem is jointly non-convex with respect to decision variables. Nevertheless, the problem is individually convex with respect to $\tau^c_{k,j}$ and jointly convex with respect to $\{\mu^c_{k,j}, S_{k,j}, \gamma_j\}$. Also, the constraints are independently separable in terms of $\tau^c_{k,j}$ and other decision variables. In other words, the minimization of KL_{mix} of the overall GMM in terms of the constrained mean and covariances is equivalent to minimizing the KL divergence between a pair of Gaussian modes. Hence, we propose a two step algorithm, where, at each step, we solve a convex optimization problem that can be solved for global optimality. In the first step, we solve for the individual components of the Gaussian mixture by minimizing the KLD of Gaussian distribution. The corresponding optimization problem is given by (4.53) - (4.56). Therefore, N_c conic optimization problems have to be solved in this step. Let us denote the objective function values of the optimization problems in the first step as $D_{KL}(q_j||r_j) = D^j_{KL}$. In the second step, we solve for $\tau^c_{k,j}$ by formulating the following optimization problem:

$$\min_{\tau^c_{k,j}} \quad D_{KL}(q||r) = \sum_{j=1}^{N_c} \tau^c_{k,j} \Big(D^j_{KL} + \log \frac{\tau^c_{k,j}}{\tau^a_{k,j}} \Big) \tag{4.69}$$

$$\text{s.t.} \quad 0 \leq \tau^c_{k,j} \leq 1; \quad j = 1, \ldots, N_c \tag{4.70}$$

$$\sum_{j=1}^{N_c} \tau^c_{k,j} = 1 \tag{4.71}$$

where the second term in the objective function signifies the relative entropy of component weights, and it is convex. Therefore, this optimization problem is convex and hence can be solved for globally optimality.

Illustration

Let $r(x)$ be a two-dimensional bi-modal Gaussian mixture distribution. Its parameters are given as $\mu_{k,1} = [10; 30]; \mu_{k,2} = [20; 31]; P_{k,1} = P_{k,2} = [4, 2; 2, 4]$. The aim of this example is to illustrate that minimizing the KLD of the mixture distribution is equivalent to minimizing the KLD of the individual components. Furthermore, we demonstrate that the optimal values of the parameters $\{\mu^c_{k,1}, \mu^c_{k,2}, P^c_{k,1}, P^c_{k,2}\}$ are independent of the choice of $\tau^c_{k,j}$.

Now, let us set $\tau^c_{k,j} = [0.7; 0.3]$ and solves for $\{\mu^c_{k,1}, \mu^c_{k,2}, P^c_{k,1}, P^c_{k,2}\}$ by minimizing the KLD between the r_1, q_1 and r_2, q_2. Figure 4.2 shows the variation of KLD of the mixture and KLD of the first component of the mixture with respect to μ_k. It can be seen that the optimal value of μ_k obtained by minimizing the KLD of the mixture is same as the optimal value obtained by minimizing the individual components of the KL of the mixture. Similar observation can be made for $\tau^c_{k,j} = [0.4; 0.6]$. Also, it can be inferred that the choice of $\tau^c_{k,j}$ does not alter the optimal values of μ_k. Therefore, optimizing the KLD of the mixture distribution is equivalent to optimizing the KLD of the individual components. Figure 4.4 shows the KL divergence with varying values of $\tau^c_{k,1}$.

Theorem 4.3: The mixture component weights of the constrained posterior distribution are equal to the corresponding mixture component weights of the unconstrained distribution

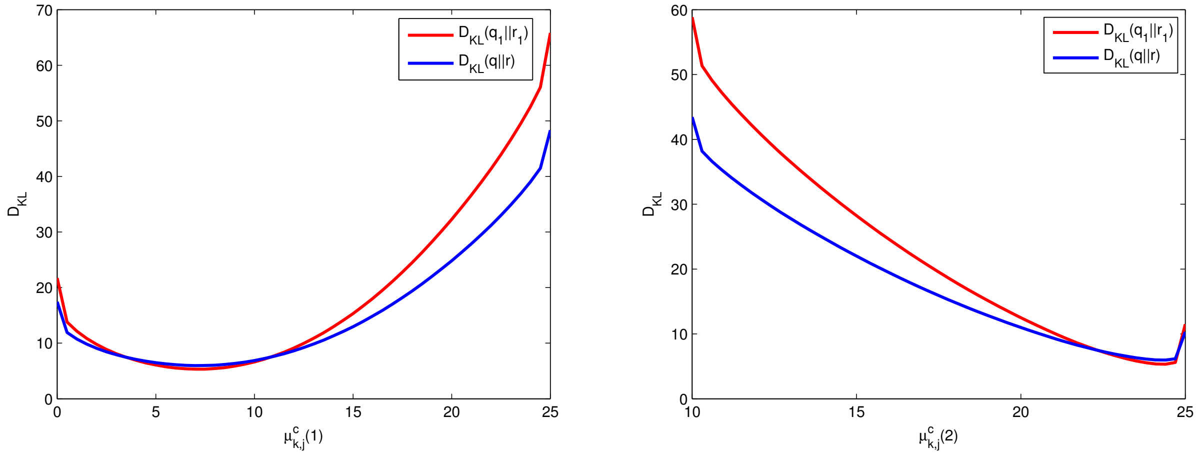


Figure 4.2: KLD between (r_1, q_1) and (r, q) versus $\mu^c_{k,1}$ when $\tau^c_{k,j}$ takes value of $[0.7; 0.3]$.

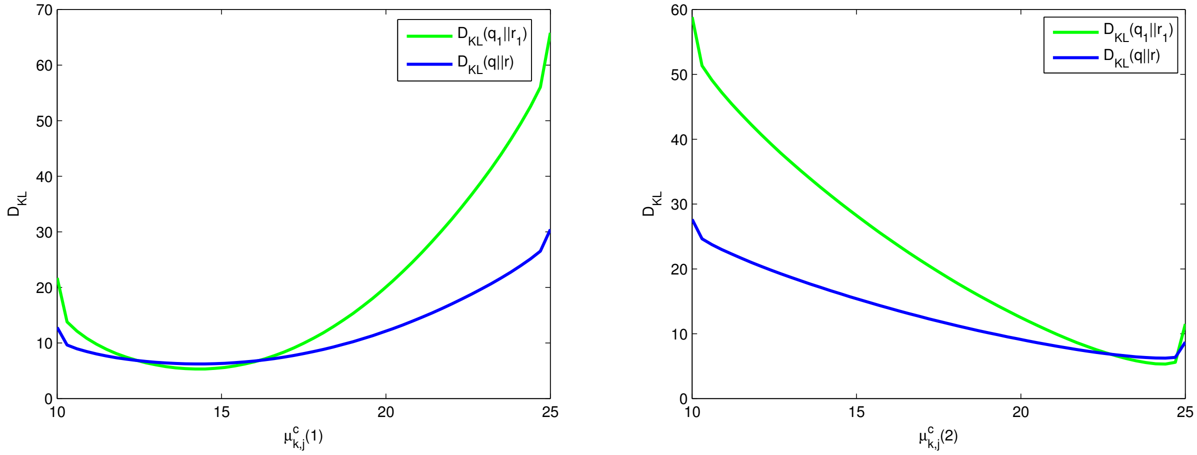


Figure 4.3: KLD between (r_1, q_1) and (r, q) versus $\mu^c_{k,1}$ when $\tau^c_{k,j}$ takes value of $[0.4; 0.6]$.

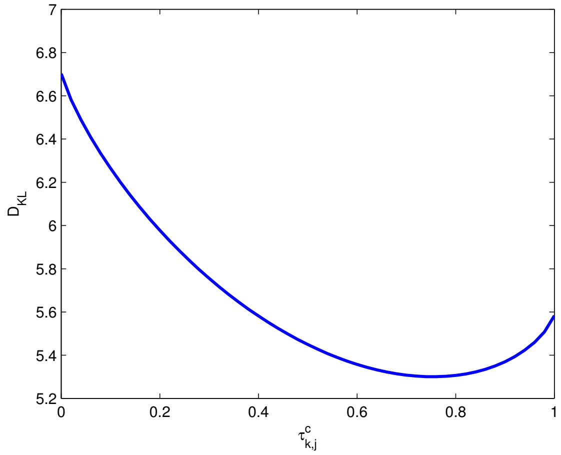


Figure 4.4: D_{KL} between $r(x)$ and $q(x)$ versus $\tau^c_{k,j}$.

($\tau^c_{k,j} = \tau^a_{k,j}$), if the KL divergence between each pair of individual components is equal (i.e., $D_{KL}(q_1||r_1) = D_{KL}(q_2||r_2) = \cdots = D_{KL}(q_{N_c}||r_{N_c})$).

Proof: After solving Eq.(4.63) for $\mu^c_{k,j}$, $P^c_{k,j}$,$j = 1, \cdots, N_c$, the KL divergences between individual modes are known, which are denoted as D^j_{KL}. The equation for optimization in terms of $\tau^c_{k,j}$ described in Eq.(4.69) can be written as:

$$\frac{\partial D_{KL}}{\partial \tau^c_{k,j}} = D^i_{KL} + \log \tau^c_{k,j} - \log \tau^a_{k,j} + 1 = 0 \tag{4.72}$$

$$\log \tau^c_{k,j} = \log \tau^a_{k,j} - 1 - D^j_{KL} \quad j = 1, \cdots, N_c \tag{4.73}$$

Given that $\sum_{j=1}^{N_c} \tau^c_{k,j} = 1$ and $\sum_{j=1}^{N_c} \tau^a_{k,j} = 1$, we can replace a certain $\tau^c_{k,d}$ with $1 - \sum_{j \neq d} \tau^c_{k,j}$, and $\tau^a_{k,j}$ with $1 - \sum_{j \neq d} \tau^a_{k,j}$. We have:

$$\log(1 - \sum_{j \neq d} \tau^c_{k,j}) = \log(1 - \sum_{j \neq d} \tau^a_{k,j}) - D^d_{KL} - 1 \tag{4.74}$$

Subtracting the jth ($j = 1, \ldots, N_c, j \neq d$) equation of Eq.(4.73) with Eq.(4.74), we have:

$$\log \frac{\tau^c_{k,j}}{1 - \sum_{j \neq d} \tau^c_{k,j}} = \log \frac{\tau^a_{k,j}}{1 - \sum_{i \neq d} \tau^a_{k,j}} - (D^d_{KL} - D^j_{KL}) \quad j = 1, \ldots, N_c, \quad j \neq d \tag{4.75}$$

If $D^1_{KL} = \cdots = D^j_{KL} (j = 1, \ldots, N_c, j \neq d)$, Eq.(4.75) becomes:

$$\log \frac{\tau^c_{k,j}}{1 - \sum_{j \neq d} \tau^c_{k,j}} = \log \frac{\tau^a_{k,j}}{1 - \sum_{i \neq d} \tau^a_{k,j}} \tag{4.76}$$

$$\tau^c_{k,i}(1 - \sum_{i \neq d} \tau^a_{k,j}) = \tau^a_{k,j}(1 - \sum_{j \neq d} \tau^c_{k,j}) \quad j = 1, \ldots, N_c, \quad j \neq d$$

Adding the above (N_c-1) equations yields:

$$\sum_{j\neq d}\tau^c_{k,j}(1-\sum_{j\neq d}\tau^a_{k,j}))=\sum_{i\neq d}\tau^a_{k,j}(1-\sum_{j\neq d}\tau^c_{k,j}) \tag{4.77}$$

$$\sum_{j\neq d}\tau^c_{k,j}=\sum_{j\neq d}\tau^a_{k,j} \tag{4.78}$$

From Eq.(4.78), we have:

$$\tau^c_{k,d}=\tau^a_{k,d} \tag{4.79}$$

Repeating this process for each $q_j, j=1,\cdots,N_c$, we have $\tau^c_{k,j}=\tau^a_{k,j}, j=1,\cdots,N_c$. □

4.6 Case study: Lorenz model

The Lorenz model was first developed by a meteorologist Edward Lorenz when he attempted to build a simplified model for convection of the earth's atmosphere. It is a classical example of highly nonlinear, non-periodic system and paradigmatic low-dimensional chaotic system. Therefore it is ideal to serve as a benchmark to test our GMM-EnKF algorithm as it will provide a good source for multimodality of the state distribution. The Lorenz model is described by the following three nonlinear ODE [62]:

$$\begin{aligned}\frac{dx}{dt}&=\gamma(y-x),\\ \frac{dy}{dt}&=\rho z-y-xz,\\ \frac{dz}{dt}&=xy-\beta z\end{aligned}$$

where the commonly used values for the coefficients are $\gamma=10,\rho=28,\beta=8/3$.

Behavior of the Lorenz model

Figure 4.5 depicts the trajectory of the projection of the Lorenz attractor onto the xz plane with the initial state values as $(x_0,y_0,z_0)=(1.508870,-1.531271,25.46071)$ within a simulation period of $[0,10]$. The initial ensemble of 1000 particles are generated from a Gaussian distribution with mean (x_0, y_0, z_0) and covariance as $diag(1, 1, 1)$. Figure 4.6 shows the evolution of these 1000 particles at time 0.2, 0.3 and 0.4 projected onto the xz plane. In the figure, the blue cluster shows the shape of the distribution of the particles. The distribution of the state deviates further and further from the shape of an ellipse. Both figure 4.5 and figure 4.6 show the strong nonlinearity of Lorenz model . Therefore it is severely inaccurate to use just Gaussian distribution to approximate the state PDF.

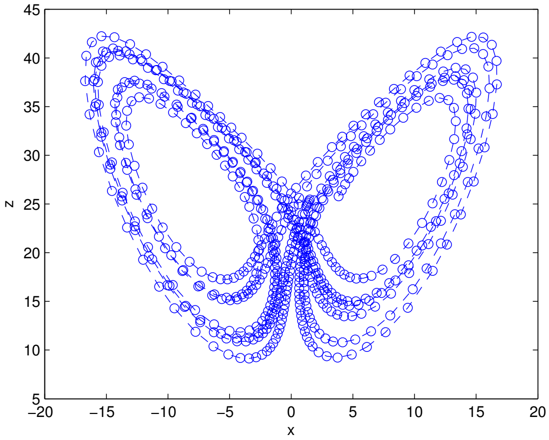


Figure 4.5: The Lorenz attractor along the xz plane.

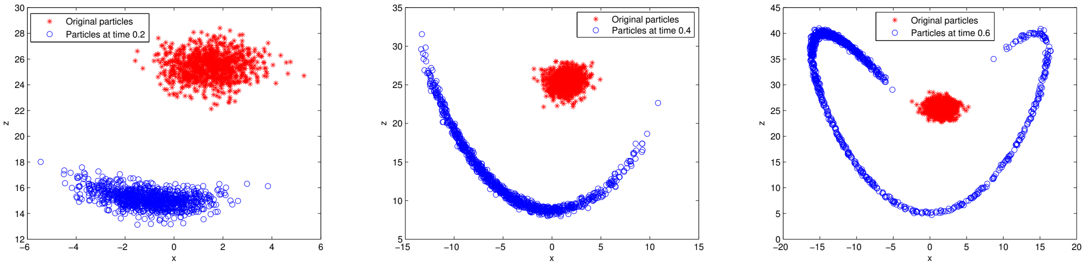


Figure 4.6: Distribution of particles at time initial time(red dots) and time step 0.2, 0.3 and 0.4(blue dots) along the xz plane.

Unconstrained data assimilation

To perform data assimilation for the Lorenz model, the measurement model is assumed to be:

$$y = [x, y, z] \tag{4.80}$$

Assume no noise for the state transition model and the measurement noise is Gaussian distribution with zero mean and covariance of $v = diag(40, 40, 40)$. In the following, we perform a one-step data assimilation using the GMM-EnKF described in the previous chapter for the Lorenz model at time steps $t_1 = 0.2, t_2 = 0.3, t_3 = 0.4$ respectively. Accordingly the measurements at these three time instances are $y_1 = [0, 0, 15], y_2 = [-2.2, -3.9, 11.9], y_3 = [-5.5, -10, 11.5]$.

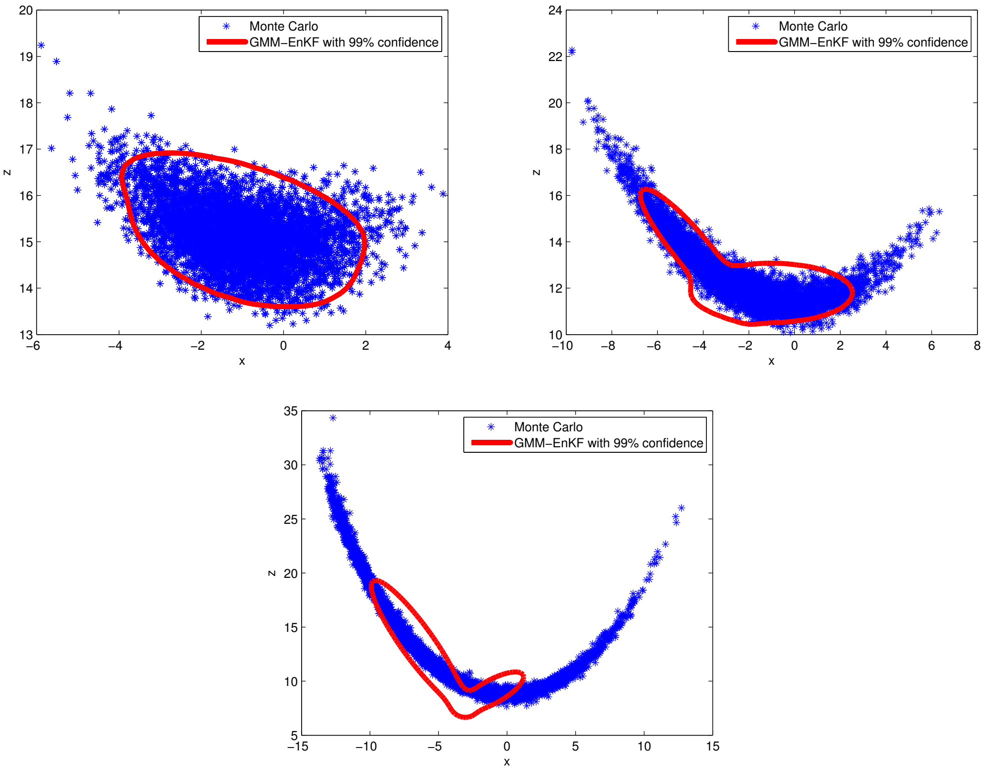


Figure 4.7: Distribution of particles using Monte Carlo method and posterior distribution using GMM-EnKF at time 0.2, 0.3 and 0.4(from the first figure to the third) along xz plane.

The red shape in figure 4.7 shows a 99% confidence region of the estimated bi-modal posterior distribution using the GMM-EnKF at time 0.2, 0.3 and 0.4. On the other hand, a Monte Carlo simulation is performed to get a closest estimation of the true distribution with 5000 particles. The blue clusters in figure 4.7 show the distribution of the particles using the Monte Carlo method at time instance 0.2, 0.3 and 0.4. The contour of the cluster can be used as the reference of the true distribution. At time 0.2 and 0.3(the top two figures in figure 4.7), the confidence region of the GMM-EnKF shown in red color can basically enclose all the particles, which means the estimated distribution is accurate at this stage. At time 0.3, the posterior distribution provided by the GMM-EnKF can well capture the multimodality of the true distribution. However, the estimation at time 0.4(bottom figure in 4.7) is not that as good as earlier, which is a result of the increasing nonlinearity of the Lorenz model. One cause is the number of the modes in GMM is fixed to be two in the estimation. Increasing this number can achieve better performance. Moreover, GMM itself is not ideal for clustering for data set with non-convex shape.

Constrained data assimilation

In this section, we consider imposing the following three-dimensional equality constraint on the states $[x, y, z]$:

$$\begin{bmatrix} -7 \\ -10 \\ 5 \end{bmatrix} \leq \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 16 \end{bmatrix}$$

The inequality constraints are incorporated using the KL divergence based method at every time instance. In the following, we present the one-step estimation of the constrained estimation results when the simulation interval is 0.2, 0.3 and 0.4.

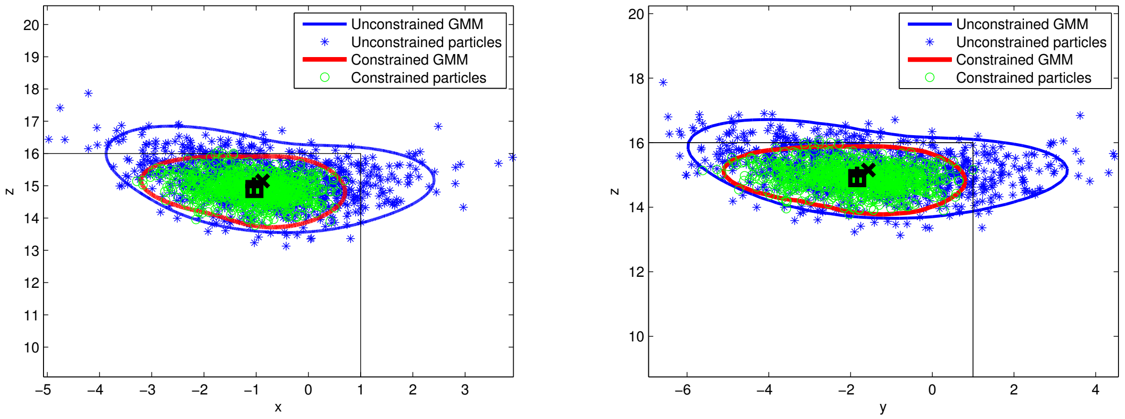


Figure 4.8: Unconstrained and constrained (using KLD approach) posterior distribution and particles at time 0.2 along (1)xz plane and (2)yz plane. The markers '+','×','square' represent the true state value, unconstrained estimate and constrained estimate.

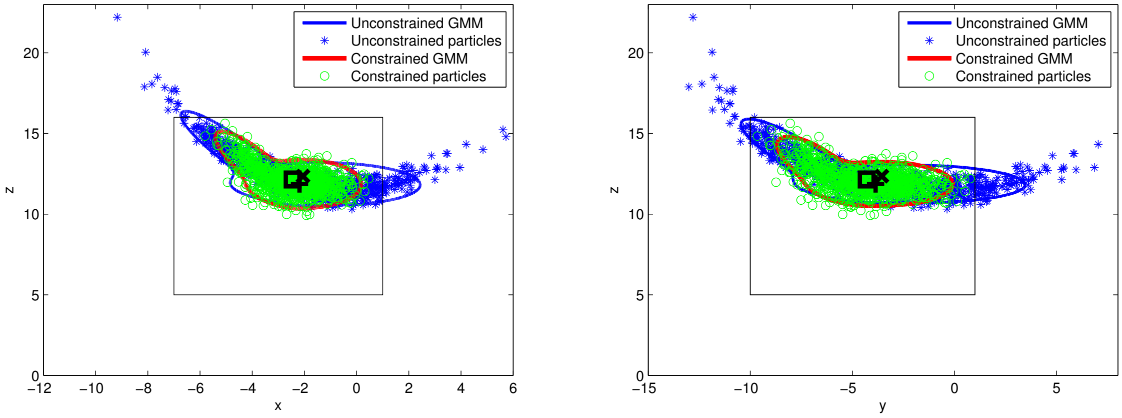


Figure 4.9: Unconstrained and constrained (using KLD approach) posterior distribution and particles at time 0.3 along (1)xz plane and (2)yz plane. The markers '+','×','square' represent the true state value, unconstrained estimate and constrained estimate.

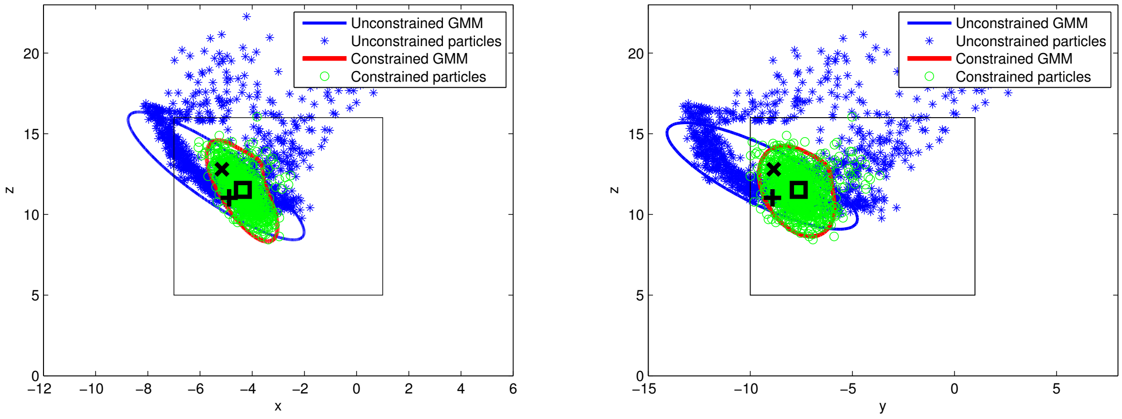


Figure 4.10: Unconstrained and constrained (using KLD approach) posterior distribution and particles at time 0.4 along (1)xz plane and (2)yz plane. The markers '+','×','square' represent the true state value, unconstrained estimate and constrained estimate.

Figure 4.8 to figure 4.10 show the 99% contour of the unconstrained posterior distribution $f(x_{k|k})$ and unconstrained particles $\{x_k^{a,i}\}_{i=1,\cdots,N}$ obtained with GMM-EnKF and 99% contour of the constrained posterior distribution $g(x)$ and constrained particles $\{x_k^{c,i}\}_{i=1,\cdots,N}$ on xz and yz plane at time 0.2, 0.3 and 0.4. As we can see from these three figures, the proposed constraining approach produces a new GMM, denoted as the red shape, which lies within the constrained region with a user defined confidence. As the parameters of the red shape have been calculated in the constraining step, this constrained distribution with analytic expression is used for further processing. The clipping approach described before usually produces a PDF with piece-wise analytic description. The RNDDR, on the hand, only produces a bunch of scattered particles.

Table 4.1 show the point estimate error of the unconstrained and constrained estimation at time 0.2, 0.3 and 0.4 in comparison with the true state value. We can see that the unconstrained estimate, in this case, is good. No constraints are required for the when we apply the GMM-EnKF on the Lorenz model. The inequality constraints described above are manually chosen only to test the effectiveness of our proposed constraining method. However, with the constraint information added, most particles are pulled back into the constraint, hence the estimates get even better.

Since the GMM-EnKF is also a particle based approach, the RNDDR approach can also be applied to achieve a constrained solution. Specifically, the quadratic optimization described in Eq.(4.81) is formulated on each of the particle in the ensemble. As we introduced in the previous chapter, the RNDDR projects the particles into the constrained region based on a weighted combination errors of states and measurements. Figure 4.11 to figure 4.13 show the distribution of constrained particles using the RNDDR approach in comparison with the unconstrained ones from the GMM-EnKF.

$$\underset{x_k^{c,i}}{min}(y - h(x_k^{c,i}))^T R^{-1}(y - h(x_k^{c,i})) + (x_k^{c,i} - x_k^{a,i})^T P_k^{-1}(x_k^{c,i} - x_k^{a,i}) \tag{4.81}$$

$$s.t. \quad lb \leq x_k^{c,i} \leq ub$$

In this particular case, the measurement error covariance R is much larger than the state covariance P_k, therefore the first term in Eq. (4.81) is much smaller than the second term, Without the correction of the measurements, many constrained particles end up lying on the boundary of the constrained region. This creates large estimation error if a large number of the constrained particles lie on the boundary, because these particles do not improve the estimation as they all take the boundary value. Reducing the measurement error covariance R for RNDDR will improve the estimation error. The KLD based method, on the other hand, is able to re-distribute the particles evenly on the newly calculated constrained GMM, so that the variability of the particles will not be compromised in the most possible way. Table 4.1 lists the point estimate error using the KLD approach and the RNDDR approach. We can see that the estimation error of RNDDR at time 0.1 and 0.3 is relatively larger than the KLD approach, because at these two steps a number of the unconstrained particles lie outside the constraints and the they are projected on the boundary in the RNDDR. The KLD, on the other hand, can push the particles more inside, closer to the true value. At time step 0.2, most unconstrained particles are already within the constraints. Therefore fewer particles are on the boundary. However, in a special case where the true value is very close to the boundary, there is no definitive conclusion on which of KLD or RNDDR renders better constrained estimation results.

From computation efficiency point of view, another advantage of KLD method is that it takes less than half of the computation time of the RNDDR approach, shown in Table 4.2. This is because it only performs one optimization at every time step instead of N optimization for each of particle.

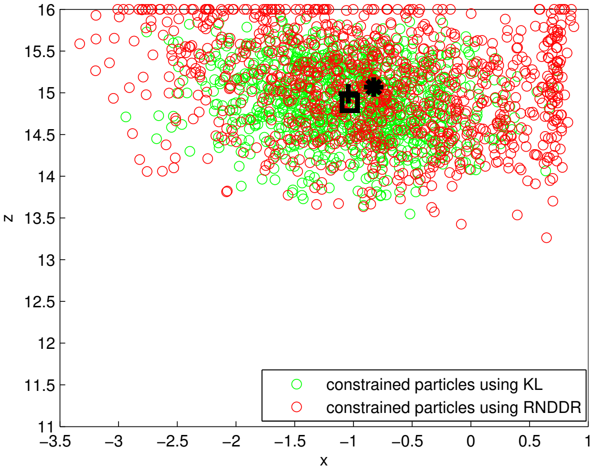


Figure 4.11: Constrained particles using the KLD and RNDDR projection at time 0.2 along xz plane. The markers '×','∗' and 'square' represent the true state value, constrained estimate using RNDDR and constrained estimate using KL.

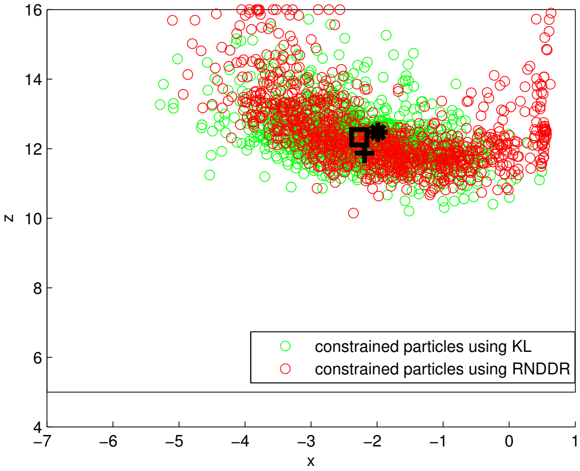


Figure 4.12: Constrained particles using the KLD and RNDDR projection at time 0.3 along xz plane. The markers '×','∗' and 'square' represent the true state value, constrained estimate using RNDDR and constrained estimate using KL.

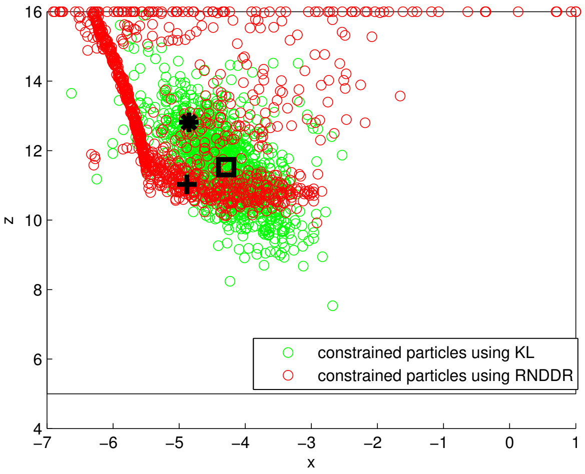


Figure 4.13: Constrained particles using the KLD and RNDDR projection at time 0.4 along xz plane. The markers '$\times$','$*$' and 'square' represent the true state value, constrained estimate using RNDDR and constrained estimate using KLD.

Table 4.1: Unconstrained estimates and constrained estimates using the KLD divergence and RNDDR

Time instance	t=0.2	t=0.3	t=0.4
Error of unconstrained estimate	0.15	0.77	3.04
Error of constrained estimate using KLD	0.03	0.18	2.85
Error of constrained estimate using RNDDR	0.19	0.25	4.19

Table 4.2: Computational Time using the KLD divergence and RNDDR

	KL approach	RNDDR approach
Average CPU Time(s)	33.2	72.6

4.7 Case study: Microalgae cultivation process

Nowadays global warming and climate change caused by increased carbon-dioxide (CO_2) emission has drawn more attention. Various carbon capture or storage technologies have been developed to reduce the CO_2 emission. Among them, the use of microalgae to biologically fixate CO_2 can be a very promising solution. Specifically, the microalgae is cultivated in a medium with all necessary nutrients, under suitable environmental conditions for microalgae to grow. It consumes CO_2 through the photosynthesis process. The complete process is environmental friendly, considering the basic cultivation medium is wastewater and no requirement of agricultural land to achieve high biomass production.

Like any other process, to achieve good performance and high efficiency of the microalgal cultivation, a mathematical model is needed to describe the complex biochemical reactions and it should also be suitable for control and optimization. Various kinetic models for microalgal cultivation process has been developed. In this work, we use a nonlinear dynamic model developed in [49] and [48]. This model is developed specifically to describe the relationship between algal growth and the CO_2, phosphate, nitrate and ammonium uptake rates of Chlorella kessleri that is indigenous to oil sand process water (OSPW). In order to generate data to establish the model, fed-batch experiments are conducted in a closed raceway photo-bioreactor with the OSPW as the cultivation medium. In the photo-bioreactor, the Chlorella kessleri is cultivated in various flasks with different initial CO_2 concentration, phosphate concentration and light intensity. The bioreactor monitors the algal biomass, concentrations of all compositions in the medium, including phosphate, ammonium, dissolved CO_2, PH and alkalinity, as well as gas content and light intensity of all flasks over 432 hours. After acquiring the data, the parameters of the model are then estimated by minimizing the weighted sum of squared error (WSSE) of the model predicted value of the states and the experimental data. In the following, we directly introduce the differential equations that describe the overall algae model. Then the estimated values of the parameters of interest are listed in Table 4.3. The mathematical model of the microalgae cultivation process can be given by the following set of equations:

$$\frac{dX}{dt} = \mu X - k_d X - DX \tag{4.82}$$

$$\frac{d[CO_2]}{dt} = -\rho_{dCO_2}X + \frac{J}{RT}\frac{h}{V}(\frac{P^{In}_{CO_2}}{h} - [dCO_2])\{1 - exp[-(\frac{k_L a_B RT}{h}\tau)]\} \tag{4.83}$$
$$+k_l\frac{S_H}{V}(\frac{P^{Out}_{CO_2}}{h} - [dCO_2]) + l[HCO_3^-][H^+] - k[dCO_2] + \frac{Q^{In}_1}{V}[dCO_2]^{In} - D[dCO_2]$$

$$\frac{d[PO_4^{-3}]}{dt} = \rho_{PO_4^{-3}}X + \frac{Q^{In}_1}{V}[PO_4^{-3}]^{In}_1 + \frac{Q^{In}_2}{V}[PO_4^{-3}]^{In}_2 - D[PO_4^{-3}] \tag{4.84}$$

$$\frac{d[NH_4^+]}{dt} = \rho_{NH_4^+}X + \frac{Q^{In}_1}{V}[NH_4^+]^{In}_1 - D[NH_4^+] \tag{4.85}$$

where the four states X, $[CO_2]$, $[PO_4^{-3}]$ and $[NH_4^+]$ are the concentrations of biomass, CO_2, phosphate and ammonium. A multi-rate sampling is performed for the process: the biomass, phosphate and ammonium are sampled every 4 h, and the CO_2 is sampled every 5 min. We get a fully observed model every 4h and only one observation every 5 min. The observability matrix of the model at the 5 min interval shows that it is unobservable, therefore, we focus on the model at the 4 h interval in this work. We measure all the four states of the process and hence the measurement model is given by:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ CO_2 \\ PO_4^{-3} \\ NH_4^+ \end{bmatrix}$$

In [49], a detailed model of the algae growth rate μ, CO_2 uptake rate ρ_{dCO_2}, phosphate uptake rate $\rho_{PO_4^{-3}}$, and ammonium uptake rate $\rho_{NH_4^+}$ are further introduced as additional

Table 4.3: The estimated parameters of the algae model

Parameter	Estimated Value	Unit	Parameter	Estimated Value	Unit
μ_m	3.89×10^{-2}	1/h	k_d	1.13×10^{-3}	$1/h$
K_{SdCO_2}	2.69	mg_{dCO_2}/L	K_{IdCO_2}	3.27×10^3	mg_{dCO_2}/L
$K_{SPO_4^{-3}}$	101	$mg_{PO_4^{-3}}/L$	$K_{IPO_4^{-3}}$	2.2×10^4	$mg_{PO_4^{-3}}$
K_{Sl}	29.1	$\mu mol\ photonsm^{-2}s^{-1}$	k_r	7.28	$\mu mol\ photonsm^{-2}s^{-1}$
K_{SdO_2}	$4.57 \times 10{-}2$	mg_{dO_2}/L			
ρ_{mdCO_2}	1.73×10^{-3}	$mg_{dCO_2}/(mgh)$	$K_{\rho CO_2}$	0.939	mg_{dCO_2}/L
$\rho_{mPO_4^{-3}}$	2.2×10^{-5}	$mg_{PO_4^{-3}}/(mgh)$	$K_{\rho PO_4^{-3}}$	831	$mg_{PO_4^{-3}}/L$
$\rho_{mNH_4^+}$	0.342	$mg_{NH_4^+}/(mgh)$	$K_{^+_4}$	3.58×10^3	$mg_{NH_4^+}/L$
k_L	0.263	m/h	$a_B\tau$	0.836	h/m
k	0.356	1/h	l	2.38×10^3	$L/(molh)$

states along with its associated parameters. Of these parameters, we are interested in the algal specific growth rate μ, which can be defined as:

$$\mu(t) = \mu_m \times \mu_{CO_2} \times \mu_{PO_4^{-3}} \times \mu_l$$

where μ_{CO_2}, $\mu_{PO_4^{-3}}$ and μ_l are the factors influenced by CO_2, phosphate concentration and light intensity. μ_m is the maximum growth rate of the algae, which is an inherent property of the algae to describe the rate of one algae strain dividing into daughter strains. So far most of the algal models assume all algae strains have the same or similar growth rate. To estimate the growth rate using Bayesian filters, the Gaussian distribution is sufficient to describe the growth rate in this case. However, it is well known that the the algae strains can divide into variable number of daughter strains, especially under different environmental conditions, such as day light. In such case, the value of the growth rate might vary and hence the distribution of the growth rate might be multimodal.

To seek evidence for this conjecture, a joint state and parameter estimation is performed on the four states and the maximum growth rate μ_m using the GMM-EnKF and the standard EnKF. The μ_m is treated as an augmented state and estimated together with the four states. Both filters start with the same initial condition and run with process noise with zero mean and covariance as $diag(1,1,1,1)$ and measurement noise with zero mean and covariance $diag(4,4,4,4)$. An ensemble of equal number of particles is used by both filters. Therefore the only variable in the estimation of the two filters is the number of modes in the augmented state distribution of every time step. The EnKF apparently has only one mode for the state PDF and the GMM-EnKF is assumed to have two. In the end, by comparing the sum of squared error (SSE) of the point estimates at all time steps yielded by each filter, we can determine if the assumption of bi-modality for the augmented state distribution stands.

Case 1: The initial distribution for both EnKF and GMM-EnKF is a Gaussian distribution with mean as $[60, 50, 350, 25]$ and covariance as $diag(25, 25, 25, 25)$ for the four states. We treat the values of parameters in table 4.3 as their true values except μ_m since it is assumed unknown and to be estimated. Nonetheless the value of $\mu_m = 3.89 \times 10^{-2} 1/h = 64.8 \times 10^{-5} 1/min$ in table 4.3 can still be a good reference for the true value of μ_m. We are only concerned about the coefficient part in the μ_m and add the exponent part 10^{-5} after we obtain the estimation. In the following, all the values associated with μ_m state only the coefficient part. The actual value has to be multiplied by 10^{-5}. The initial distribution of μ_m is also assumed to be Gaussian for both filters, with mean 80 and variance 100. Figure 4.14 compares the estimated values of the four states over 432h using the GMM-EnKF and the EnKF respectively. Note that because the sampling time is 4h, there are 109 time steps in total. Figure 4.15 compares estimated values of the μ_m over the whole time using the two filters respectively. The black line in the figure shows the reference value of μ_m from table 4.3.

Case 2: Since the unconstrained estimations of the four concentrations do not produce nonphysical estimation results over all time steps and also track the trajectory of the experimental data well, there is no need to put constraints on the four states. However, the estimation of the μ_m in figure 4.15 yields non-physical results, for the reason that μ_m cannot be negative. In this case, an inequality constraint is imposed just on the parameter μ_m, specifically $30 < \mu_m < 100$(without the exponent part). The incorporation of the constraints

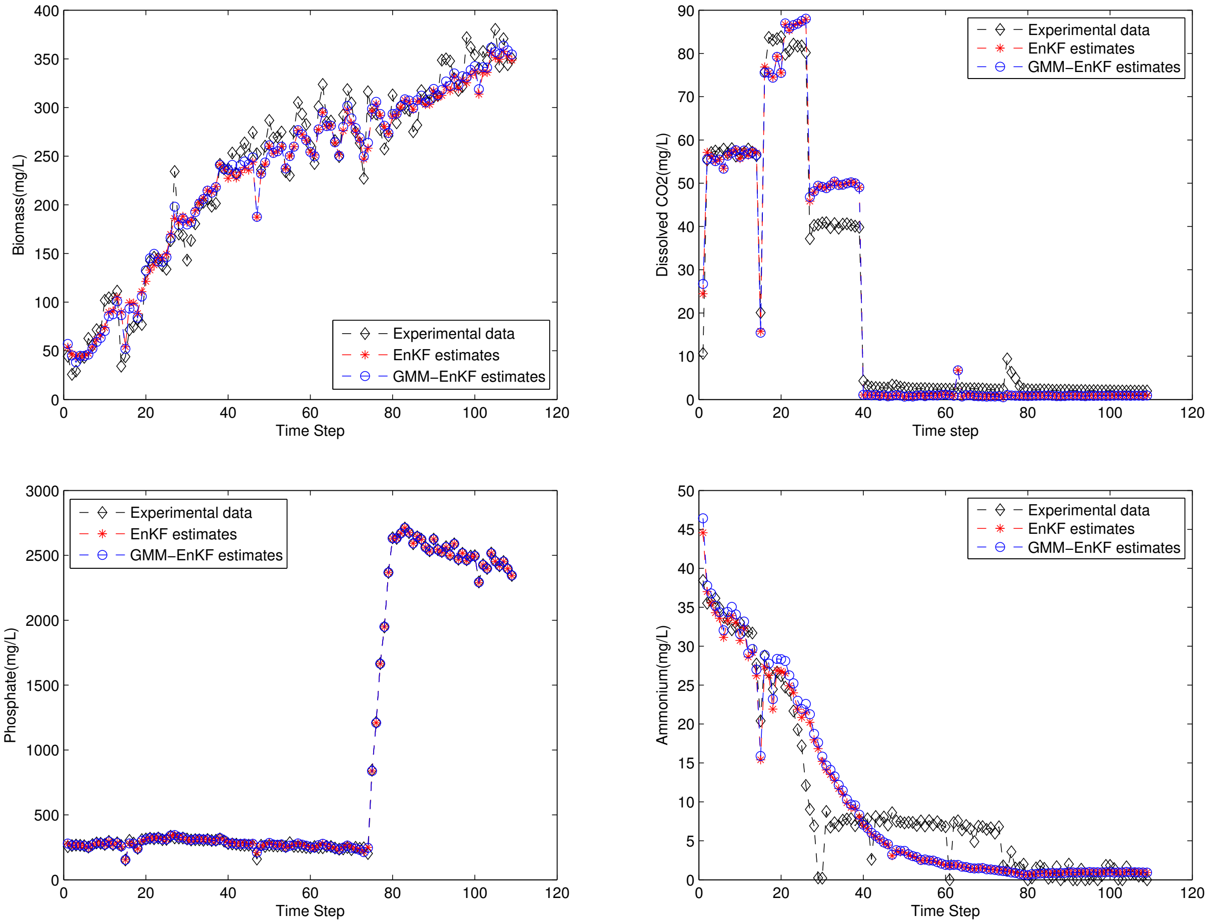


Figure 4.14: Experimental data and estimation results of the four states using the EnKF and GMM-EnKF over 432h.

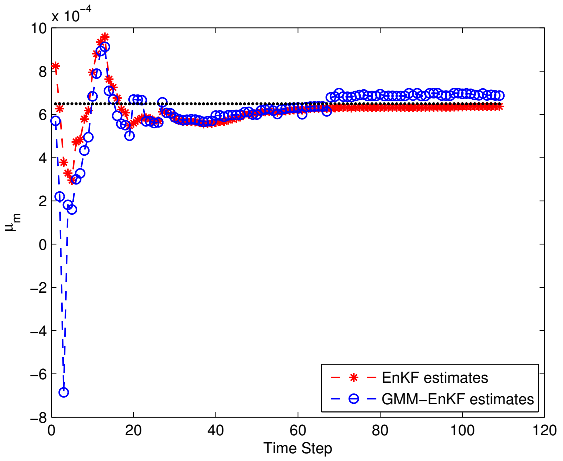


Figure 4.15: Comparison of estimated μ_m over 432h using:the EnKF and GMM-EnKF

on μ_m is achieved by using the proposed KLD based method. Figure 4.16 shows the comparison of estimation results of the four states using the GMM-EnKF and the constrained GMM-EnKF. The two filters have similar performance because unconstrained results in most time steps fall within the constraints, hence the constraints are not active in many time steps for the constrained estimation. Nevertheless improvement is still visible, especially in the biomass estimation in figure 4.16. Figure 4.17 shows the constrained estimation of μ_m in comparison with the unconstrained version and the EnKF.

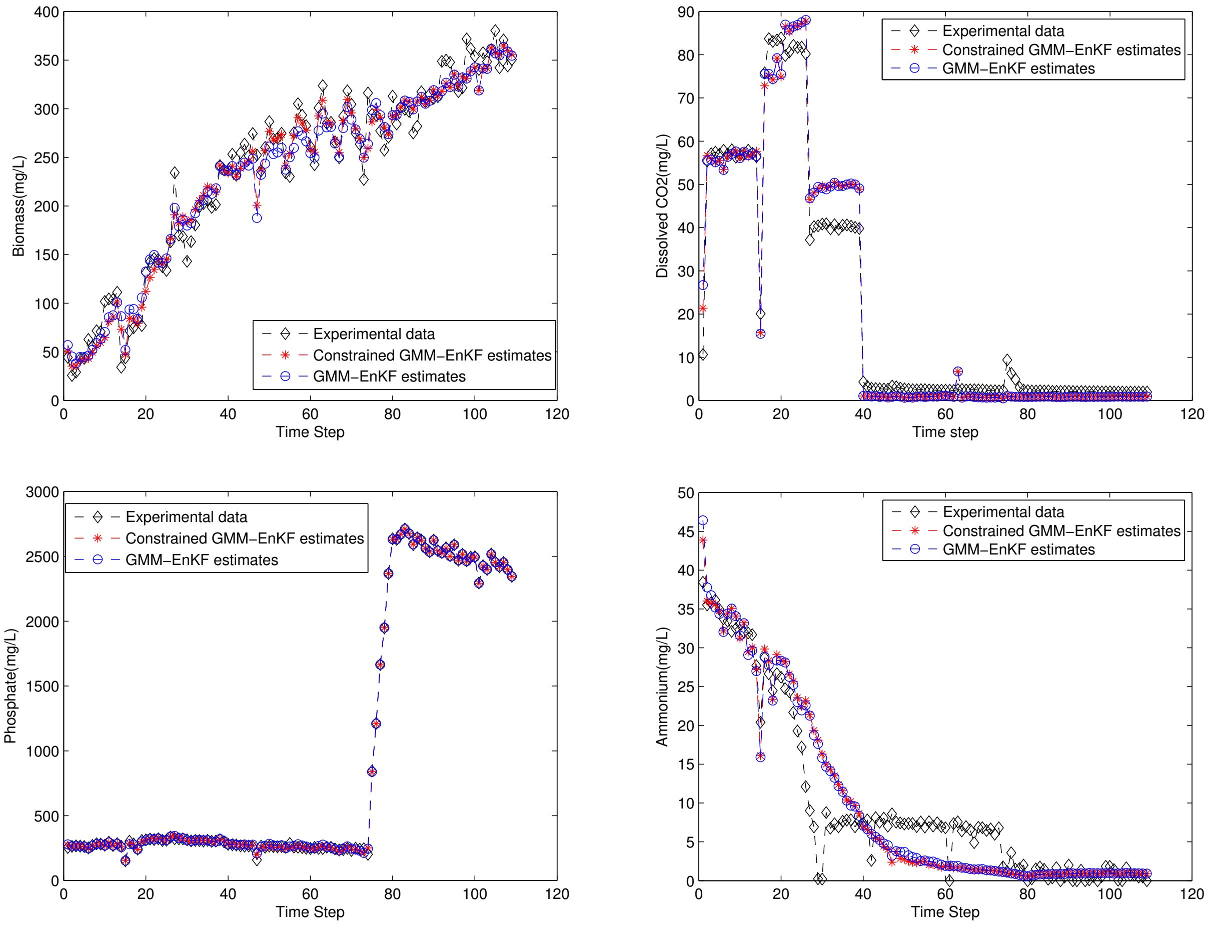


Figure 4.16: Experimental data and estimation results of the four states using the GMM-EnKF and constrained GMM-EnKF over 432h.

Analysis:

Table 4.4 lists the sum of square of the error (SSE) of the four states generated by the EnKF, GMM-EnKF and constrained GMM-EnKF respectively. The converged value of μ_m and the

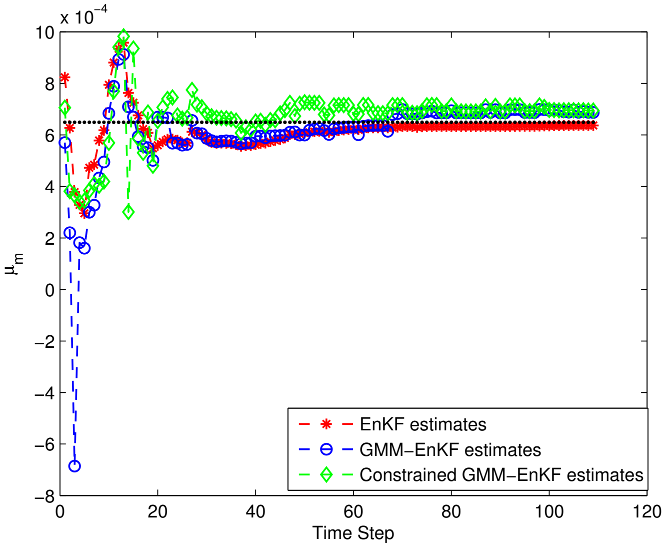


Figure 4.17: Comparison of estimated μ_m using the EnKF, the GMM-EnKF and constrained GMM-EnKF over 432h.

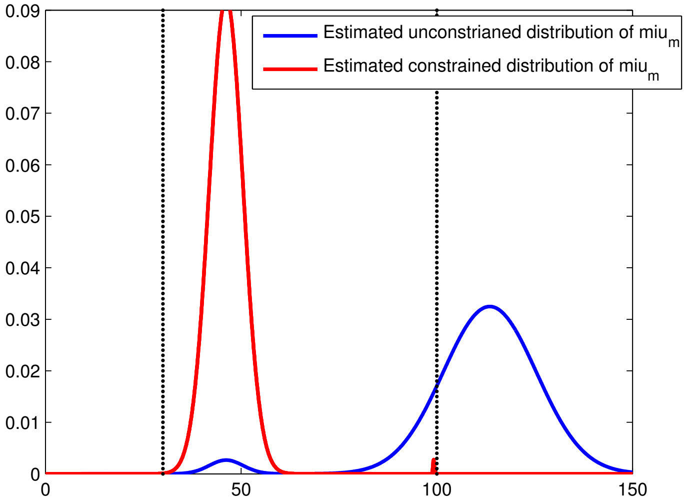


Figure 4.18: The estimated distribution of the μ_m before and after enforcing the constraint using KL method at one time step(area between the two dotted lines is the constrained area).

time steps it takes to converge under each scenario are also listed in table 4.4. From the figures and Table 4.4, we have the following findings:

(1) The fact that the GMM-EnKF outperforms the EnKF indicates that there is a strong possibility that the distribution of μ_m is multi-modal, i.e. it can take different values in the whole cultivation process. Again the GMM-EnKF proves a better option than the EnKF in the presence of multi-modal PDF.

(2) It is expected that the constrained GMM-EnKF outperforms all the unconstrained filters simply because extra information added to the estimation, which also helps the GMM-EnKF converge faster than the unconstrained filters. For the parameter estimation, the constrained GMM-EnKF is to get rid of the non-physical estimations. However, we see small oscillations even after the estimation converges. This is because of the mode-seeking behavior of the reverse KLD introduced in chapter 3. That's to say the reverse KLD tends to have the peaks of the constrained distribution trace the peaks of the unconstrained distribution. As is shown in figure 4.18, obviously one of the unconstrained modes has most part outside the constraints. The constrained result keeps the one that is inside the constraint, and disregards the other one by putting very little weight on it. By doing this, the constrained result is shifted drastically from the unconstrained result, so are the particles. This large shifting of the particles at every time step is the major reason of the oscillation of constrained GMM-EnKF.

(3) In the constraining step, the quadratic term $(y-h(x))R^{-1}(y-h(x))^T$, where x is the to-be-determined constrained estimate, is used to employ the measurement to help correct the constrained estimate. However, the measurement model $h(.)$ cannot reflect estimated parameter to the measurement. In other words, the measurement at the current time step cannot be used to correct the constrained estimate. Only in the next step can the measurements be used to correct the constrained update of the parameter. The measurement information is able to be incorporated in the update of the constrained state estimation because $h(.)$ can pass the state information to the measurements at the current time step.

This is another reason why there are small oscillations for the constrained results.

Table 4.4: The estimated parameters of the algae model

	EnKF	GMM-EnKF(Case 1)	Constrained GMM-EnKF
X(SSE)	4.359×10^4	3.604×10^4	3.09×10^4
CO_2(SSE)	0.207×10^4	0.205×10^4	0.203×10^4
PO_4(SSE)	1.38×10^4	1.307×10^4	0.967×10^4
NH_4(SSE)	0.181×10^4	0.199×10^4	0.191×10^4
Converged μ_m value($\times 10^{-5}$)	63.68	68.70	69.45
Converged time(steps)	60	60	45

4.8 Conclusions

In this chapter, we propose a novel approach to incorporate inequality constraints into a non-Gaussian filter that characterizes the posterior using a Gaussian mixture model in the EnKF filtering framework. The idea behind our approach is to project the unconstrained posterior GMM distribution into the constrained region such that the resulting GMM lies within the constraints. The parameters of the constrained GMM is obtained by minimizing the KL divergence between the two GMMs. As there is no close-form expression for the KL divergence between two GMMs, we proposed a modification of Goldberger's approximation as the objective function of the proposed optimization problem at the update step. Since the resulting approximation is non-convex, we proposed a two step optimization procedure that can determine the posterior distribution of the constrained filter. Each step requires one to solve convex optimization problems that can be solved for global optimality. This enables one to generate ensemble members to be inside the constraints and hence provide better state estimates. Furthermore, we compare the performance of our proposed KLD based approach with the existing RNDDR approach, and we show superior performance of the proposed approach over the RNDDR when the measurement error is large. In this case, the particles generated by RNDDR tend to lie on the boundary of constraints, which might create great error unless the true value happens to be very close to the boundary. On the other hand, in our approach, we can always keep the particles inside the constraints, closer to the true value. In addition, compared to the RNDDR, we are able to provide a parametric model for the constrained posterior distribution. Finally, our method is computationally more efficient than the RNDDR.

Chapter 5

Gaussian Mixture Model-Based Ensemble Kalman Smoother

5.1 Introduction

As Monte Carlo based filtering techniques such as the EnKF provide a convenient and efficient way to handle processes with nonlinearity and high-dimensionality, it is natural to extend their application to smoothing. Ensemble smoothing(ES) [100] performs one global update of each particle in the ensemble by simultaneously assimilating all data available. Although the ES saves computation time significantly compared to the EnKF, the EnKF outperforms the ES when the model is strongly nonlinear, because the recursive framework of the EnKF keeps the estimation at every time step on track [22] for nonlinear dynamics. The ensemble Kalman smoother(EnKS)[25] is able to improve the performance of ES by adapting it into a recursive update structure. The EnKS allows us to update the past estimate with future observations in the same fashion as the EnKF. Therefore, the EnKS inherits the EnKF's ability to handle nonlinearity and high-dimensionality. In [25], the EnKF, the ES and the EnKS are applied on the Lorenz model which is well-known for its nonlinear and chaotic dynamics. The EnKS shows clearly significant improvement of the estimation error compared to the EnKF and the ES. Even the EnKF is shown to be superior to the ES in this example, which demonstrates the weakness of the ES with nonlinear models, especially those with chaotic dynamics. However, [33] shows that the ES and the EnKS have close performance on a weakly nonlinear model - a fish stock assessment model. The ES also gives better estimation results in diffusive systems compared to chaotic systems, such as reservoir models [89].

The most widely used formulation of EnKS is a fix-lagged smoother proposed by Evensen et al. [25]. In this formulation, the observation acquired at time step k is used not only to conduct filtering for time step k, but also to perform smoothing on a previous time step $k - L$, where L is a fixed integer. As the extension of the EnKF, the EnKS operates based upon the EnKF filtering results. First, the EnKF runs from $k - L$ to k and the ensemble at the prediction step as well as the update step are both collected and stored at every time step. Second, for every upcoming measurement $y_k(k > L)$, the EnKS updates x_{k-L} again using the smoothing formulation. As this window of length of L moves forward, the updated time steps keep moving outside the window to make space for the next time step. This lagged smoother is able to save CPU memory space since the storage for the ensemble of those time steps which move outside the window can be released.

The ES, the EnKF and the EnKS are all based on the assumption that the statistics of the state space can be approximated by Gaussian distributions. As is introduced in chapter 2, this assumption does not hold when the smoother is applied on strongly nonlinear models, which is frequently encountered in ocean or meteorologic applications. The motivation of our work is to obtain the full distribution of the non-Gaussian state space for the smoother at every time step, instead of just the first two moments. The variational data assimilation technique[61][58] has been widely used by the weather forecast community for high-dimensional models with non-Gaussian state statistics. However, methods in this category usually involve complicated optimization of a cost function. We intend to deal with the non-Gaussian problem in the sequential data assimilation framework. To the best of our knowledge, very few publications have focused on the topic of sequential non-Gaussian smoother. [11] proposed an iterative ensemble Kalman smoother(IEnKS), which is a hybrid method to combine the EnKF with the four-dimensional ensemble variational method(4DVAR). Another approach to handle non-Gaussian PDF is sequential Monte Carlo smoothing, i.e. the particle smoother[18]. The SMC smoothing algorithms can be performed in two different formulations. The first is the RTS smoother formulation, i.e. forward filtering and backward smoothing algorithm. The second is to run two individual filters in two opposite directions: one assimilates data forward and the other backward. In the end, the filtering results of the two filters are combined to generate the smoothing results. In [40], the two smoothers are applied on a linear system. The estimation shows the two have similar performance, with the two-filter smoother operating faster.

In the previous chapter, a GMM-EnKF has been introduced and tested for its capability to handle the non-Gaussian state probability density function(PDF). We intend to extend its formulation to a smoother, so that this smoother would share its ability to handle the non-Gaussian PDF. As this smoother also approximates the state PDF with GMM, we name it GMM-EnKS in the following content. In our work, the proposed GMM-EnKS adopts the RTS smoother formulation, in which the forward filtering is performed by the GMM-EnKF. We also proposed a novel method to update the probability matrix in the backward smoothing, by adapting the idea of the particle smoother to the GMM-EnKS.

5.2 The Ensemble Kalman Smoother (EnKS)

In this section, two variants of the ensemble Kalman smoothers are introduced in details. The first one is the EnKS introduced by Evensen et al. [25] which can be formulated as a fixed-lag or fixed-interval smoother. It intends to provide the smoothed estimates at a certain time step k over an interval $[1 \quad T]$ or $[T-L \quad T]$ by calculating the joint PDF of the states given observations from time step 1 to T, i.e. $p(x_{0:T}|y_{1:T})$ (fixed-interval smoother) or

$p(x_{T-L:T}|y_{1:T})$(fixed-lag smoother). The integer L is the lag. The second one is the ensemble formulation of the Rauch-Tung-Striebel(RTS) smoother(EnRTS) [93], which is used less commonly than the EnKS. The EnRTS intends to find the marginal PDF of the states at a certain time step k over an time interval $[1 \quad T]$, given observations from 1 to T, i.e. $p(x_k|y_{1:T})$. In [75], the EnKS and the EnRTS are proven to be equivalent, even in the nonlinear, non-Gaussian scenario.

5.2.1 The EnKS

Consider the following discrete model:

$$x_{k+1} = f(x_k) + v_k \tag{5.1}$$

$$y_k = Hx_k + w_k \tag{5.2}$$

where the the process noise v_k and measurement noise w_k are both Gaussian and given by $v_k \sim N(0, Q)$, $w_k \sim N(0, R)$. As is introduced in the previous chapter, the EnKF updates each individual particle x_k^i of the ensemble as follows.

$$x^{a,i}_{k|k} = x^{f,i}_{k|k-1} + K(Y^{obs}_k - Hx^{f,i}_{k|k-1} + w_k) \tag{5.3}$$

where $x^{f,i}_{k|k-1} = f(x^i_{k-1|k-1}) + v_k$. The Kalman gain $\mathbf{K}$ is given by:

$$K = C_{xy}(C_{yy} + R)^{-1} \tag{5.4}$$

$$C_{xy} = \frac{1}{N-1} X'_{k|k-1} Y'^{T}_{k|k-1} \tag{5.5}$$

$$C_{yy} = \frac{1}{N-1} Y'_{k|k-1} Y'^{T}_{k|k-1} \tag{5.6}$$

where the superscript prime indicates that the matrix has its mean removed from each column.

When the observation Y_T^{obs} at time step T is available, the EnKS updates all the previous states in the time interval $[T-L \quad T]$, where $1 \leq L \leq T$. In the following, we focus on the fixed-interval smoother. When $k=1$, the EnKS is initialized with the EnKF analysis results, $x^s_{1|1} = x^a_{1|1}$. At time step k, the analysis of the smoother to assimilate the future observation of time step T, $T \geq k+1$, is performed as follows:

$$x^{s,i}_{k|T} = x^{s,i}_{k|T-1} + B(Y_T^{obs} - Hx^{f,i}_{T|T-1} + w_k) \tag{5.7}$$

where B is the smoothing gain and is given by:

$$B = C_{xy}(C_{yy} + R)^{-1} \tag{5.8}$$

$$C_{xy} = \frac{1}{N-1} X'_{k|T-1} Y'^{T}_{T|T-1} \tag{5.9}$$

$$C_{yy} = \frac{1}{N-1} Y'_{T|t-1} Y'^{T}_{T|t-1} \tag{5.10}$$

where $X'_{k|T-1}$ is calculated based on the ensemble $\{x^i_{k|T-1}\}_{i=1,\ldots,N}$ and Y'_T is calculated based on the ensemble $\{y^i_{T|T-1}\}_{i=1,\ldots,N}$, which are the corresponding measurements of the predicted ensemble $\{x^i_{T|T-1}\}_{i=1,\ldots,N}$ obtained from the filtering at time step T.

If the smoothing process starts at time step 1, when Y_2^{obs} is available, the ensemble at time step 1 is updated using Eq.(5.7) as follows:

$$x^{s,i}_{1|2} = x^{s,i}_{1|1} + B(Y_2^{obs} - Hx^{f,i}_{2|1} + w_2) \tag{5.11}$$

where $x^{s,i}_{1|1}$ is the assigned with the filtering values. Note that in the following content, the variable with the superscript s indicates it is the smoothed estimate.

C_{xy} and C_{yy} are calculated based on $\{x^i_{1|1}\}_{i=1,...,N}$ and $\{y^i_{2|1}\}_{i=1,...,N}$ as follows:

$$C_{xy} = \frac{1}{N-1} X'_{1|1} Y'^T_{2|1} \tag{5.12}$$

$$C_{yy} = \frac{1}{N-1} Y'^T_{2|1} Y'^T_{2|1} \tag{5.13}$$

When Y^{obs}_3 arrives, the smoothing can be performed on $x_{1|3}$ and $x_{2|3}$. The calculation of $x_{2|3}$ is similar to Eq.(5.11) and $x_{1|3}$ is calculated as:

$$x^{s,i}_{1|3} = x^{s,i}_{1|2} + B(Y^{obs}_3 - Hx^{f,i}_{3|2} + w_3) \tag{5.14}$$

where C_{xy} and C_{yy} are calculated based on ensemble $\{x^{s,i}_{1|2}\}_{i=1,...,N}$, which comes from the smoothing of last time step and $\{y^i_{3|2}\}$, which come from the filtering step. The same procedure is is repeated for future observations and we have all the smoothing estimates from 1 to T.

By comparing Eq.(5.7) with Eq.(5.3), the most prominent difference of the EnKS from the EnKF is that the calculation of the cross-covariance matrix C_{xy} involves the ensembles from two different time steps, time step k at which the smoothing is performed and time step T at which the observation arrives. The error covariance of measurement C_{yy} is calculated the same way as in the filtering step. Moreover, the smoothing does not require any more computation than the EnKF, since there are no extra forward model runs required in the EnKS. The ensemble information required can be either obtained from the filtering step or the previous smoothing steps.

5.2.2 The RTS EnKS

The EnKS introduced in the last section is a filter-smoother approach, i.e. after it performs the filtering, the smoothing is conducted retrospectively within the fixed interval or lag determined beforehand. Compared to the EnKS, the EnRTS is structured in a forward filtering and backward smoothing formulation, which is the same as the the linear RTS Gaussian smoother introduced in chapter 1. For a time interval, it first performs the filtering in the forward direction and the smoothing in the backward direction. The EnRTS is carried out in the following manner.

1. The forward filtering pass

The forward pass is performed by the EnKF, which calculates the posterior estimate $x_{k|k}$ at every time step k of the interval $[0 \quad T]$ using the formulation described in Eq.(5.3). The predicted ensemble $\{x^{f,i}_{k|k-1}\}_{i=1,\ldots,N}$ and the updated ensemble $\{x^{a,i}_{k|k}\}_{i=1,\ldots,N}$ in the forward pass are stored for use in the backward smoothing pass.

2. The backward smoothing pass

The backward pass starts at time step T, at which time the initial smoothing ensemble is assigned to be equal to the updated ensemble, $\{x^{s,i}_{T|T}\} = \{x^{a,i}_{T|T}\}$. Then the smoothing is performed retrospectively from $k = T$ to $k = 1$, during which the ensemble at time step k is updated as follows:

$$x^{s,i}_{k|T} = x^{a,i}_{k|k} + B(x^{s,i}_{k+1|T} - x^{f,i}_{k+1|k}) \tag{5.15}$$

where $\{x^{a,i}_{k|k}\}$ are the updated ensemble of filtering at time step k and $\{x^{f,i}_{k+1|k}\}$ are the predicted ensemble of the filtering at $k + 1$. $\{x^{s,i}_{k+1|T}\}$ is the smoothed ensemble of $k + 1$, which is the smoothed ensemble of the last smoothing time step since the smoothing is performed retrospectively. **B** is the smoothing gain which is given by:

$$B = C_{xx}[P_{xx}]^{-1} \tag{5.16}$$

$$C_{xx} = \frac{1}{N-1} X^{'}_{k|k} X^{'T}_{k+1|k} \tag{5.17}$$

$$P_{xx} = \frac{1}{N-1} X^{'}_{k+1|k} X^{'}_{k+1|k} \tag{5.18}$$

Again the prime matrix indicates the mean has been removed from each column. Eq.(5.17) and Eq.(5.18) indicate that $C_{xx} = cov(\{x^{a,i}_{k|k}\}, \{x^{f,i}_{k+1|k}\})$ and $P_{xx} = cov(\{x^{f,i}_{k+1|k}\}, \{x^{f,i}_{k+1|k}\})$, where *cov* represents the covariance.

As is seen from Eq.(5.16) to Eq.(5.18), same as the EnKS, the EnRTS does not require any forward model runs either because the ensemble information required either comes from the filtering or the previous smoothing. However, the EnRTS consumes much more CPU storage than the EnKS because it needs to store the predicted and updated ensemble of every time step of the filtering in the whole interval $[0\ \ T]$ before the backward smoothing starts. For such reason, it is usually regarded as an off-line smoother. The EnKS can be implemented online by limiting the length of the interval. As the smoothing does not need to be waited before the filtering is completely done for the interval, the storage required for smoothing based on previous observations can be released when the new observation arrives.

Just like the EnKF, both the EnKS and the EnRTS are subject to the assumptions that the prior and posterior distributions are Gaussian. Violation of this assumption will lead to less accurate estimation results.

5.3 The Gaussian mixture model ensemble Kalman smoother (GMM-EnKS)

5.3.1 Problem statement

In this work, for a nonlinear system we attempt to estimate the marginal state PDF at time step k given all observations available in the interval $[0\ \ T]$, which is:

$$p(x_k|y_{1:T})$$

The forward-backward RTS smoothing framework is adopted here for estimation. In the RTS approach, using the Bayesian inference and Markov properties, the smoothed PDF can be written as:

$$p(x_k|y_{1:T}) = p(x_k|y_{1:k}) \int \frac{p(x_{k+1}|x_k)}{p(x_{k+1}|y_{1:k})} p(x_{k+1}|y_{1:T}) dx_{k+1} \tag{5.19}$$

where $p(x_k|y_{1:k})$ is the posterior PDF of the filtering and $p(x_{k+1}|y_{1:T})$ is the last-step posterior PDF of the smoothing.

Since we consider estimating the posterior PDF of the filtering $p(x_k|y_{1:k})$ to be non-Gaussian, $p(x_k|y_{1:T})$ is also non-Gaussian. We approximate this posterior PDF of the smoothing with GMM, which is given by:

$$p(x_k|y_{1:T}) = \sum_{i=1}^{N_c} \tau_k^s N(x_k; \mu_k^s, P_k^s) \tag{5.20}$$

The goal of smoothing is to estimate the parameters τ_k^s, μ_k^s and P_k^s by assimilating all the available observations. Because of the non-Gaussianity, Eq.(5.19) is not tractable. Therefore, a semi-parametric smoother is proposed in this chapter to extend the EnKS to estimate the non-Gaussian state PDF, referred to as the GMM-EnKS in the following content. In this proposed smoother, the GMM-EnKF introduced in chapter 2 plays the role in the forward filtering step.

5.3.2 Revisit of the GMM-EnKF algorithm

The details of the GMM-EnKF algorithm have been introduced in chapter 2. For the purpose of clarification of the notation, a brief summary of the GMM-EnKF is introduced here. The GMM-EnKF also works based on an ensemble of particles $\{x_k^i\}, i = 1, \ldots, N$, where k is the time step. The core idea of this algorithm to handle multimodality is to attach the membership weights w_{ij}^k to all the particles in the ensemble so that the ensemble can be updated under each individual mode $j(j = 1, \ldots, M)$ of the GMM using the standard EnKF. Finally, the M updated ensembles $x_{k,j}^{a,i}$ are combined to one ensemble $x_k^{a,i}$ using the membership weights w_{ij}^k.

The GMM-EnKF consists of the prediction step and update step.

Prediction:

In the prediction step, the EM clustering is performed on the predicted ensemble $\{x_k^{f,i}\}_{i=1,\cdots,N}$

to structure the predicted distribution $p(x_k^f)$.

$$p(x_k^f) = \sum_{j=1}^{M} \tau_j^f N(x_k^f; \mu_j^f, P_j^f) \tag{5.21}$$

where M is the number of modes in the Gaussian mixture, which is pre-determined information.

Along with the parameters of $p(x_k^f)$, the membership matrix W, which indicates the probability of each particle belonging to each mode, is also calculated. W is a $N \times M$ matrix and its each element is given as follows:

$$w_{ij} = \frac{\tau_j^t N(x_i; \mu_j^t, P_j^t)}{\sum_{m=1}^{M} \tau_m^t N(x_i; \mu_m^t, P_m^t)} \tag{5.22}$$

where τ_j^t, μ_j^t and P_j^t are the estimated parameters at t step of the EM algorithm.

Update:

For each component j, the error covariance is computed with the integration of the probability matrix W.

$$C_{xy}[j] = \sum_{i=1}^{N} w_{ij}^k (x_k^{f,i} - \mu_{j,k})(Hx_k^{f,i} - H\mu_{j,k})^T / n_{j,k} \tag{5.23}$$

$$C_{yy}[j] = \sum_{i=1}^{N} w_{ij}^k (Hx_k^{f,i} - H\mu_{j,k})(Hx_k^{f,i} - H\mu_{j,k})^T / n_{j,k} \tag{5.24}$$

where $n_{j,k} = \sum_{i=1}^{N} w_{ij}^k$. The Kalman gain associated with mode j is calculated as follows:

$$K[j] = C_{xy}[j](C_{yy}[j] + R)^{-1} \tag{5.25}$$

Assuming the Gaussian component j claims ownership of all the ensemble members, the Kalman update is conducted under each component j as follows. The step gives $N \times M$

particles in the end, which are given by:

$$x_{j,k}^{a,i} = x_k^{f,i} + K[j](d_k - Hx_k^{f,i}) \tag{5.26}$$

The posterior particles are generated by combining the $N \times M$ particles to form $N \times 1$ particles by using the probability matrix W as follows.

$$x_k^{a,i} = \sum_{j=1}^{M} w_{ij}^k x_{j,k}^{a,i} \tag{5.27}$$

The updated weight, mean and covariance of each component are calculated as follows:

$$\mu_{j,k}^a = \sum_{i=1}^{N} w_{ij}^k x_{j,k}^{a,i} / n_{j,k} \tag{5.28}$$

$$P_{j,k}^a = \sum_{i=1}^{N} w_{ij}^k (x_{j,k}^{a,i} - \mu_{j,k}^a)(x_{j,k}^{a,i} - \mu_{j,k}^a)^T / n_{j,k} \tag{5.29}$$

$$\tau_{j,k}^a = p(\mu_{j,k}^a, P_{j,k}^a, R|d) = \frac{p(d|\mu_{j,k}^a, P_{j,k}^a, R)n_j}{\sum_{m=1}^{M} p(d|\mu_{j,k}^a, P_{j,k}^a, R)n_j} \tag{5.30}$$

In the forward filtering, the following variables should be stored at every time step for the backward smoothing:

(1) The updated ensemble under each mode $\{x_{j,k}^{a,i}\}$ and the final updated ensemble $\{x_k^{a,i}\}$.

(2) The predicted mean $\mu_{j,k}^f$ and updated mean $\mu_{j,k}^a$ of each mode.

(3) Membership matrix W which contains the membership weights w_{ij}^k of each particle.

5.3.3 The GMM-EnKS algorithm

After the GMM-EnKF completes the forward filtering step, the GMM-EnKS starts to conduct the backward smoothing. The idea of the proposed GMM-EnKS is to extend the exiting RTS EnKS to the multimodal PDF by utilizing an ensemble with weighted particles.

Specifically, same as the GMM-EnKF, each particle is assigned with a weight given by the membership matrix W. The GMM-EnKS attempts to accomplish two primary goals:

(1) Calculate the smoothing ensemble $\{x_k^{s,i}\}_{i=1,\dots,N}$;

(2) Calculate the updated membership matrix W^s.

Similar to the GMM-EnKF, at every time step k of the smoothing, one mode of the mixture claims ownership of the whole filtered ensemble $x_k^{a,i}$, such that the ensemble can be processed using the EnKS for this mode. Therefore, a temporary ensemble $\{x_{j,k}^{s,i}\}_{i=1,\dots,N,j=1,\dots,M}$ for each mode j similar to Eq.(5.27) has be to calculated. Under the RTS smoothing framework, it is calculated as follows:

$$x_{j,k}^{s,i} = x_k^{a,i} + B[j](x_{k+1}^{s} - x_{k+1}^{f,i}) \tag{5.31}$$

where x_{k+1}^s is the smoothing estimate of last time and $B[j]$ is the smoothing gain for mode j.

The smoothing gain associated with mode j is given by:

$$B[j] = C_{k,k+1}^s[j][C_{k+1,k+1}^s[j]]^{-1} \tag{5.32}$$

where $C_{k,k+1}^s$ is the cross-covariance calculated using the ensembles at time step k and k+1 of the filtering; $C_{k+1,k+1}^s$ is the auto-covariance calculated using the ensemble at time step k+1 of the filtering.

The calculation of the smoothing gain, specifically the cross covariance $C_{k,k+1}^s$, involves the ensembles $x_k^{a,i}$ and $x_{k+1}^{f,i}$ of two different time steps. Since the state PDf is multimodal, even though mode j of $p(x_k^a)$ at time step k is assumed to own all particles, it is unclear which mode of $p(x_{k+1}^f)$ has the ownership correspondingly at time step $k+1$. In other words, which of the $\mu_{j,k+1}^f$ to be used for the calculation of $C_{k,k+1}$ is unknown.

The mapping of the modes between the two time steps is straightforward because there is only a single mode, as is the case in the EnKS. However, the issue arises if there are multiple modes in a distribution. If we were to process each mode of time step k individually using the EnKS, it is crucial to determine the modes in $p(x_{k+1}^f)$ to which each mode in $p(x_k^a)$ corresponds. In other words, we need to find the mapping relationship between the modes of two GMMs.

$$\mu_{j,k}^a \to \mu_{q,k+1}^f, \quad j = 1, \ldots, M, \;\; q = 1, \ldots, M \tag{5.33}$$

The filtering does not directly give this mapping information $j \to q$ because the predicted distribution $p(x_{k+1}^f)$ is obtained with the EM clustering algorithm, owing to the nonlinearity of the system. In order to obtain this mapping, we propose to first calculate a prediction of the forward propagation for each mode in the mixture $p(x_k^a)$ and then use this prediction as the reference to match each mode in the mixture $p(x_{k+1}^f)$. As the parameters of a Gaussian distribution propagating through a nonlinear model cannot be determined directly, the Monte Carlo method is used to calculate this reference mode. As are shown in Eq.(5.28) and Eq.(5.29), each mode in $p(x_k^a)$ can be represented by its temporary ensemble $x_{j,k}^{a,i}$ associated with weights w_{ij}. Using the weighted particles, the reference mode $p_j(x_k^a)$ can be calculated. With the reference mode obtained, we select the mapping mode to be the closest mode in $p_j(x_{k+1}^f)$ to the reference mode, i.e. whichever of $\hat{\mu}_{q,k+1}, q = 1, \ldots, M$ has the closest distance to $\mu_{j,k+1}^f$ is deemed to be the mapping mode for mode j of time step k. The following equation concludes the mapping procedures described above:

$$p_j(x_k^a) \overset{p_j(x_{k+1}^f)'}{\Rightarrow} p_q(x_{k+1}^f) \tag{5.34}$$

where $p_j(x_k^a)$ is the jth mode in $p(x_k^a)$ and $p_q(x_{k+1}^f)$ is its mapped mode. $p_j(x_{k+1}^f)'$ is the intermediate reference mode.

In the following, the detailed procedure of the GMM-EnKS is described.

(1) The smoothing starts in the opposite direction of the filtering. When k=1, smoothing assigns the initial estimate to be equal to the filtering estimate of time step, i.e. $x_1^s = x_T^a$.

(2) At time step k, for each mode j, propagate its temporary ensemble $x_{j,k|k}^{a,i}$ through the

model.

$$x^{f,i}_{j,k+1} = f(x^{a,i}_{j,k}) + v_{k+1} \tag{5.35}$$

(3) The mean of the reference Gaussian mode is the weighted summation of the ensemble above.

$$\mu^{'}_{j,k+1} = \sum_{i=1}^{N} w^{k}_{ij} x^{f,i}_{j,k+1} / n_{i,k} \tag{5.36}$$

where $n_{i,k}$ is the normalization term, $n_{i,k} = \sum_{i=1}^{N} w_{ij}$.

(4) For each mode $p_j(x^a_k)$, calculate the Euclidean distances of the mean of its reference mode $p_j(x^f_{k+1})^{'}$ and the mean of each predicted mode $p_q(x^f_{k+1})$.

$$d_{jq} = \parallel \mu^{f}_{q,k+1} - \mu^{'}_{j,k+1} \parallel \quad j = 1, \ldots, M, q = 1, \ldots, M \tag{5.37}$$

(5)For each mode $p_j(x^a_k)$, find q such that d_{jq} is minimum.

$$c(j) = \min_{q} d_{jq} \tag{5.38}$$

where $c(.)$ is the matching relationship between j($j = 1, \ldots, M$) and q($q = 1, \ldots, M$).

From what is described above, $\mu^{'}_{j,k+1}$ bridges $\mu^{a}_{j,k}$ and $\mu^{f}_{j,k+1}$. By finding the closest mode to this reference mode in the predicted Gaussian mixture $p(x^f_{k+1})$, we can match the mode j of time step k and mode q of time step $k + 1$. After the mapping of the modes between the two GMMs has been established, the GMM-EnKS algorithm can be performed using the EnKS.

(6) At time step k, assume the ensemble $x^{a,i}_{k}$ only belongs to one mode j. The cross-covariance of the ensemble at time step k and k+1 and auto-covariance of the ensemble at time step k+1 are calculated as follows:

$$C^{s}_{k,k+1}[j] = \sum_{i=1}^{N} w^{k}_{ij} (x^{a,i}_{k} - \mu^{a}_{j,k})(x^{f,i}_{k+1} - \mu^{f}_{c(j),k+1})^{T} / n_{j,k} \tag{5.39}$$

$$C^s_{k+1,k+1}[j] = \sum_{i=1}^{N} w_{ij}^{k+1}(x_{k+1}^{f,i} - \mu^f_{c(j),k+1})(x_{k+1}^{f,i} - \mu^f_{c(j),k+1})^T / n_{j,k+1} \tag{5.40}$$

where $\mu^f_{c(j),k+1})$ is the matching mode of $\mu^a_{j,k}$. $n_{j,k} = \sum_{i=1}^{N} w^k_{ij}$. The w^k_{ij} and w^{k+1}_{ij} represent the membership matrix of time step k and k+*1* at the filtering step.

(7) The smoothing gain associated with mode j is calculated as follows:

$$B[j] = C^s_{k,k+1}[j][C^s_{k+1,k+1}[j]]^{-1} \tag{5.41}$$

(8) The update of each particle under each component in the smoothing step is given by:

$$x^{s,i}_{j,k} = x^{a,i}_{k} + B[j](x^s_{k+1} - x^{f,i}_{k+1}) \tag{5.42}$$

where x^s_{k+1} is the estimate of the state at last time step of smoothing.

(9) Combine the $N \times M$ particles in Eq.(5.42) into $N \times 1$ particle to get the final smoothing ensemble.

$$x^{s,i}_k = \sum_{j=1}^{M} w^k_{ij} x^{s,i}_{j,k} \tag{5.43}$$

(10) The weight, mean and covariance of each component at time step k are updated as follows.

$$\mu^s_{j,k} = \sum_{i=1}^{N} w^k_{ij} x^{s,i}_{j,k} / n_{j,k} \tag{5.44}$$

$$P^s_{j,k} = \sum_{i=1}^{N} w^k_{ij}(x^{s,i}_{j,k} - \mu^s_{j,k})(x^{s,i}_{j,k} - \mu^s_{j,k})^T / n_{j,k} \tag{5.45}$$

$$\tau^s_{k,j} = \frac{p(d|\mu^s_{j,k}, P^s_{j,k}, R)n_j}{\sum_{m=1}^{M} p(d|\mu^s_{j,k}, P^s_{j,k}, R)n_{j,k}} \tag{5.46}$$

(11) The final smoothing estimate is given by:

$$x_k^s = \sum_{j=1}^{M} \tau_{k,j}^s \mu_{j,k}^s \tag{5.47}$$

The GMM-EnKS algorithm works similar to the GMM-EnKF algorithm except that the covariance $C_{k,k+1}^s[j]$ and $C_{k+1,k+1}^s[j]$ involve two Gaussian mixture distributions from two time steps, which introduces the difficulty of finding out the matching relationship between them. The pseudo-code of GMM-EnKS is provided as follows.

Algorithm 5.1: GMM-EnKS algorithm.
Inputs are the updated ensemble $\{x_k^{a,i}\}$ and mean of each mode $\mu_{j,k}^a$, the
temporary updated ensemble $\{x_{j,k}^{a,i}\}$, the predicted ensemble $\{x_{j,k+1}^{f,i}\}$
and mean of each mode $\mu_{j,k}^f$, the probability matrix W of every time step;
total filtering steps T

```
for $k = 1 : (T-1)$
  for $i = 1 : N$
    Calculate $x_{j,k+1}^{f,i} = f(x_{j,k}^{a,i}) + v_{k+1}$
    Calculate $\mu'_{j,k+1} = \sum_{i=1}^{N} w_{ij}^k x_{j,k+1}^{f,i} / \sum_{i=1}^{N}(w_{ij}^k)$
    for $i = 1 : M$
      for $j = 1 : M$
        Calculate $d_{ij} = \| \mu_{i,k+1}^f - \mu_{j,k+1}^{'} \|$
      end for
    end for
    for $i = 1 : M$
      $c(j) = \underset{i}{min} d_{ij}$
    end for
    Calculate the cross-covariance $C_{k,k+1}^s[j]$ and auto-covariance $C_{k+1,k+1}^s[j]$ using
    Eq.(5.39) and Eq.(5.40)
    Calculate the smoothing gain using Eq.(5.41)
    Calculate the temporary updated $x_{j,k}^{s,i}$ using Eq.(5.42)
    Calculate the final updated $\mu_{j,k}^s$ using Eq.(5.43)
  end for
  for $j = 1 : M$
    Calculate the weight, mean and covariance of each mode $\mu_{j,k}^s$ using
    Eq.(5.44) to Eq.(5.46)
  end for
end for
```

5.4 Update of Probability Matrix $\boldsymbol{W}$ in the GMM-EnKS

In the GMM-EnKS framework introduced in the previous section, the probability matrix W at every time step is assumed to take the same values as the one calculated in the filtering step. However, W should also be updated by assimilating the observations.

5.4.1 Revisit of importance sampling

Assume that a random variable x follows a distribution $p(x)$. Importance sampling intends to generate a set of samples $\{x^{(i)}\}_{i=1,2,\cdots,N}$ from $p(x)$, with each sample associated with a weight w_i, such that p(x) can be approximated by the following discrete representation:

$$\begin{aligned} p(x) &\approx \frac{1}{N}\sum_{i=1}^{N}\delta(x-x^{(i)}) \\ &= \sum_{i=1}^{N} w_i\delta(x-x^{(i)}) \end{aligned} \tag{5.48}$$

With this discrete approximation of p(x), the integrals described in Eq.(5.49) can be evaluated.

$$\hat{f}(x) = \int f(x)p(x)dx \tag{5.49}$$

$$\begin{aligned} \hat{f}(x) &\approx \int f(x)\sum_{i=1}^{N} w_i\delta(x-x^{(i)})dx \\ &= \sum_{i=1}^{N} w_i f(x^{(i)}) \end{aligned} \tag{5.50}$$

where $\hat{f}(x)$ can indicate the value of any moment of x. In particular, the first two moments of x can be obtained from:

$$\bar{x} = \sum_{i=1}^{N} w^{(i)}x^{(i)} \tag{5.51}$$

$$P_x = \sum_{i=1}^{N} w^{(i)}(x^{(i)} - \bar{x})(x^{(i)} - \bar{x})^T \tag{5.52}$$

In importance sampling, the samples $x^{(i)}$ are taken from an easier-to-sample distribution $q(x)$ referred to as the importance density. The corresponding weights $w^{(i)}$ are defined as:

$$w^{(i)} = \frac{p(x^{(i)})}{q(x^{(i)})} \tag{5.53}$$

It has to be pointed out that the method used to generate samples $x^{(i)}$ along with their associated weights $w^{(i)}$ is not exclusive to the one introduced above. As long as this yet-to-be-determined method can yield a batch of random weighted samples to approximate $p(x)$ with the discrete form of Eq(5.48), it is feasible to use.

5.4.2 Sequential update of importance weights in smoothing

Recall that in the prediction step of the GMM-EnKF, the parameters of the GMM are calculated in the M step of the EM algorithm as follows:

$$\tau_j^{k+1} = \frac{N^k}{N} \tag{5.54}$$

$$\mu_j^{k+1} = \frac{1}{N^k} \sum_{i=1}^{N} w_{ij} x_i \tag{5.55}$$

$$P_j^{k+1} = \frac{1}{N^k} \sum_{i=1}^{N} w_{ij}(x_i - \mu_j^{k+1})(x_i - \mu_j^{k+1})^T \tag{5.56}$$

where w_{ij} are the membership weights and $N^k = \sum_{i=1}^{N} w_{ij}$.

The moments are calculated the same way as are described in Eq.(5.51) and Eq.(5.52), which indicates that the data samples $\{x_i\}_{i=1,...,N}$ can be considered as a batch of weighted samples, with the membership probability w_{ij} calculated in the E-step as their weights. However, this weighted samples $\{x_i\}_{i=1,...,N}$ can only be considered as taken from one individual component j instead of the whole Gaussian mixture distribution. That's to say for each individual component j, x_i is associated with a set of different weights $w_{ij}, i = 1, \cdots, N, j = 1, \ldots, M$. Moreover, Eq.(5.44) and Eq.(5.45) also indicate that each mode in the posterior distribution can be represented with weighted samples $x^{s,i}_{j,k}$. The idea introduced here is to consider that the ensemble at every time step, prior or posterior, can be considered as Monte Carlo samples with different sets of weights for each component.

$$p_j(x_k) = \sum_{i=1}^{N} w^k_{ij} p_j(x^{(i)}_k) \tag{5.57}$$

$$p(x_k) = \sum_{j=1}^{M} p_j(x_k) \tag{5.58}$$

where $p(x_k)$ is the individual mode of the GMM $p(x_k)$, which is the state distribution at time step k.

With the ensemble considered as weighted Monte Carlo particles, in the smoothing, the weights can therefore be updated using the sequential importance sampling(SIS) framework. In the forward filtering, the application of the SIS is not required as the EM algorithm is performed at every time step to update the membership matrix W. We consider applying SIS to update W only in the backward smoothing. This is similar to the RTS particle smoother since its primary goal is to update the weights associated with each particle. A brief introduction of the RTS particle smoother can be found in chapter 1. In the following we introduce the mathematical details of how the update is done.

The smoothed density $p(x_k|y_{1:T})$ can be factorized as:

$$\begin{aligned} p(x_k|y_{1:T}) &= \int p(x_k|x_{k+1}, y_{1:k}) p(x_{k+1}|y_{1:T}) dx_{k+1} \\ &= \frac{p(x_k, x_{k+1}|y_{1:k})}{p(x_{k+1}|y_{1:k})} p(x_{k+1}|y_{1:T}) dx_{k+1} \\ &= p(x_k|y_{1:k}) \int \frac{p(x_{k+1}|x_k) p(x_{k+1}|y_{1:T}) dx_{k+1}}{\int p(x_{k+1}|x_k) p(x_k|y_{1:k}) dx_k} dx_{k+1} \end{aligned} \tag{5.59}$$

From Eq.5.59, the smoothed density at time step k can be formulated in terms of the

$p(x_k|y_{1:k})$ from the filtering step, the state transition density $p(x_{k+1}|x_k)$ and the smoothed density at time step $(k+1)$, $p(x_{k+1}|y_{1:T})$. The posterior density of filtering $p(x_k|y_{1:k})$ can be approximated with a set of weighted particles as:

$$p(x_k|y_{1:k}) = \sum_{i=1}^{N} w_k^i \delta_{x_k^{(i)}}(x_k) \tag{5.60}$$

Similarly we attempt to approximate the posterior density of smoothing $p(x_{k+1}|y_{1:T})$ as:

$$p(x_k|y_{1:T}) = \sum_{i=1}^{N} w_{k|T}^i \delta_{x_{k|T}^{(i)}}(x_k) \tag{5.61}$$

Substituting Eq.(5.60) and Eq.(5.61) into Eq.(5.59), we have:

$$\begin{aligned} p(x_k|y_{1:T}) &= \sum_{i=1}^{N} W_k^i [\sum_{j=1}^{N} W_{k+1|T}^j \frac{f(x_{k+1}^j|x_k^i)}{[\sum_{l=1}^{N} W_n^l f(x_{k+1}^j|x_k^l)]}] \delta_{x_k^i}(x_k) \\ &= \sum_{i=1}^{N} W_{k|T}^i \delta_{x_k^i}(x_k) \end{aligned} \tag{5.62}$$

Eq.(5.62) gives the final formulation of the posterior density using SIS framework for smoothers. From Eq.(5.62), we can easily obtain the smoothing weights at time step k as follows:

$$W_{k|T}^i = \sum_{i=1}^{N} W_k^i [\sum_{j=1}^{N} W_{k+1|T}^j \frac{f(x_{k+1}^j|x_k^i)}{[\sum_{l=1}^{N} W_n^l f(x_{k+1}^j|x_k^l)]}] \tag{5.63}$$

With Eq.(5.63), $W_{k|T}^i$ can be sequentially updated in the smoothing step using the weights from the filtering step at time same step W_k^i as well as the smoothing weights of the last time step $(k+1)$, $W_{k+1|T}^j$.

In the GMM-EnKS algorithm, the weight matrix is first updated using Eq.(5.63) for each component that claims ownership of all particles and then is applied in the calculation of the smoothing gain. The error covariance now can be updated as follows:

$$C^s_{k,k+1}[j] = \sum_{i=1}^{N} w^{ij}_{k|T}(x^{a,i}_k - \mu^a_{j,k})(x^{f,i}_{k+1} - \mu^f_{c(j),k+1})^T / n_{j,k} \tag{5.64}$$

$$C^s_{k+1,k+1}[j] = \sum_{i=1}^{N} w^{ij}_{k+1|T}(x^{f,i}_{k+1} - \mu^f_{c(j),k+1})(x^{f,i}_{k+1} - \mu^f_{c(j),k+1})^T / n_{j,k+1} \tag{5.65}$$

where $\{w^{ij}_{k+1|T}\}_{i=1,\ldots,N}$ is the jth column in $W_{k|T}$.

The initial value of $W_{k|T}$ is set to be equal to the filtered W at time step T.

5.5 Case Study: A nonlinear time series model

This nonlinear time series model described in Eq.(5.66) is used extensively for testing numerical filtering techniques. In this work, the original square linear measurement operator is replaced with a linear measurement operator in Eq.(5.67). The transition model is known to show strong nonlinearity and evident multimodality, which is ideal for the demonstration of the GMM-EnKS.

$$x_k = x_{k-1} + 25\frac{x_{k-1}}{1 + x^2_{k-1}} + 8cos(1.2k) + v_k \tag{5.66}$$

$$y_k = \frac{x_k}{20} + w_k \tag{5.67}$$

where $v_k \sim N(0, 2)$ and $w_k \sim N(0, 0.5)$. The initial distribution $x_0 \sim N(0.1, 0.5)$. The number of the components in the GMM is set to two.

The forward filtering is performed using the GMM-EnkF with 200 particles over 30 time steps, and then the backward smoothing is carried out using the proposed GMM-EnKS. The smoothing results using and without using the proposed importance weights update method are shown for comparison.

In the first scenario, the estimation results of the state x_k from GMM-EnKF filtering and

GMM-EnKS smoothing are shown in figure 5.1 in comparison with the true state trajectory. We can see from figure 5.1 the smoothing slightly improves the filtering results, but the two are close to each other.

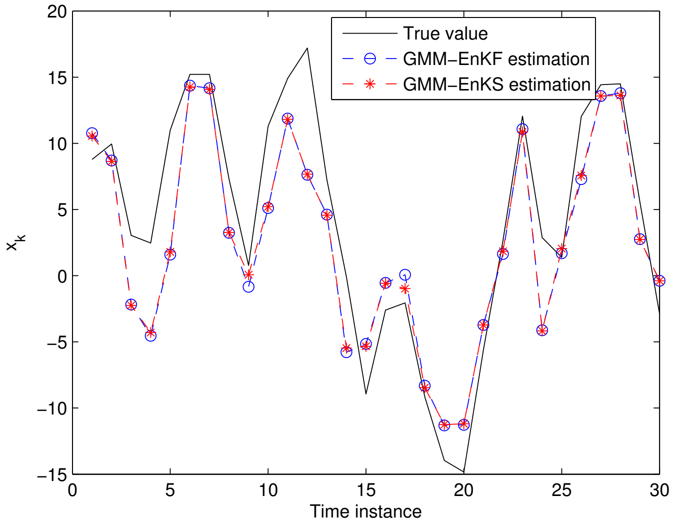


Figure 5.1: Comparison of the GMM-EnKF filtering, GMM-EnKS smoothing results and true values

In the second scenario, we perform the update on the membership weights at every time step of the smoothing using the method proposed in the previous section. Figure 5.2 shows the comparison of the smoothing results with and without using the proposed membership weights W update method. Note that by assuming no update on W, it means that W in the smoothing takes the corresponding W of the filtering step. Table 5.1 presents the summation of square error(SSE) of the GMM-EnKF, the proposed GMM-EnKS without update on W and the proposed GMM-EnKS with update on W. Table 5.1 indicates that the proposed GMM-EnKS with update on membership matrix W performs the best.

Table 5.1: Comparison of the SSE of the GMM-EnKF, GMM-EnKS and GMM-EnKS with update on W

Method	GMM-EnKF	GMM-EnKS without proposed update	GMM-EnKS with proposed update
SSE	505.3	486.2	461.7

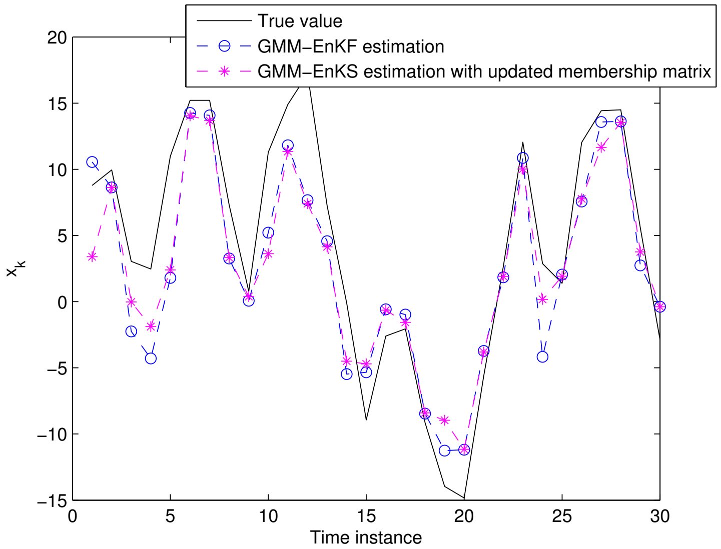


Figure 5.2: Comparison of GMM-EnKS smoothing with and without updated membership matrix respectively and true values

In the following, we use the standard EnKS with the same initial conditions and operating parameters as are used in the GMM-EnKS to test if the proposed GMM-EnKS indeed is better in dealing with non-Gaussian conditions. Figure 5.3 shows the estimation results using the EnKF and the EnKS in comparison with the true trajectory. Table 5.2 lists the SSE of the EnKF and the EnKS. Although the EnKS does improve significantly compared with the EnKF, it obviously cannot handle the strongly non-Gaussian systems as well as the GMM-EnKS. This might be true for the overall SSE of the algorithm, however not as true for each individual time step, since it improves significantly for some time steps and worsens a few at others, as is shown in figure 5.2.

Table 5.1 indicates that the GMM-EnKS improves 8.6% based on the GMM-EnKF.However, comparing Table 5.1 with Table 5.2, the proposed GMM-EnKS improves 30.9% relatively to the standard EnKS and the GMM-EnKF also improves significantly compared with the EnKF, which indicates the obvious advantage of the proposed estimators, GMM-EnKS and GMM-EnKF, when dealing with non-Gaussian systems.

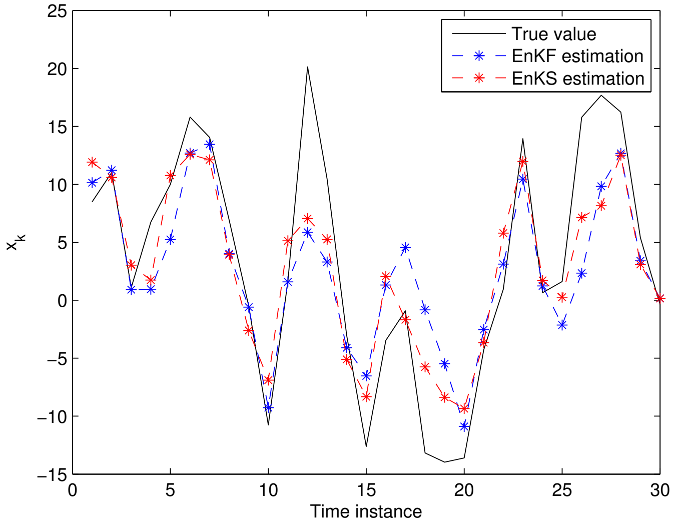


Figure 5.3: Comparison of the EnKF filtering, EnKS smoothing results and true values

Table 5.2: Comparison of the SSE of EnKF anf EnKS

Method	EnKF	EnKS
SSE	950.6	668.4

5.6 Case study: The algae cultivation process

In this case study, we apply the proposed GMM-EnKS on the algae model that has been used in chapter 4. The details of the experiment to cultivate the algae have already been introduced as well as the mathematical model for estimation application. The only difference in this case study is that only the state estimation is carried out instead of the joint state and parameter estimation as in the case study in chapter 4. The parameter maximum growth rate is assumed to be known and equal to the experimental value, which is 6.48×10^{-4}. The four states to be estimated are concentrations of the biomass X, CO_2, PO_4^{-3} and NH_4^+. The initial state distribution is Gaussian, with mean $[60, 50, 350, 25]$ and covariance as $diag(25, 25, 25, 25)$ and the process noise and measurement noise are also both Gaussian, with zero mean and covariance of $diag(1, 1, 1, 1)$ and $diag(4, 4, 4, 4)$. The process goes on for 432 hours, with CO_2 sampled every 5 minutes and the other three sampled every 4 hours. The smoothing is performed every 4h when all four states are available to measure.

Figure 5.4 shows the estimation results of the forward-filtering using the GMM-EnKF and the backward smoothing using the GMM-EnKS(without the update of the membership matrix) in comparison with the true vales of the four states respectively. Table 5.3 presents the sum of square error(SSE) of each of the four states using the GMM-EnKF and GMM-EnKS as well as the EnKF and the EnKS. Both the figure and the table show that the GMM-EnKS manages to improve further more based on the GMM-EnKF results. Moreover, the proposed GMM based estimators improve in the performance compared to their corresponding Gaussian counterparts when dealing with non-Gaussian state PDF.

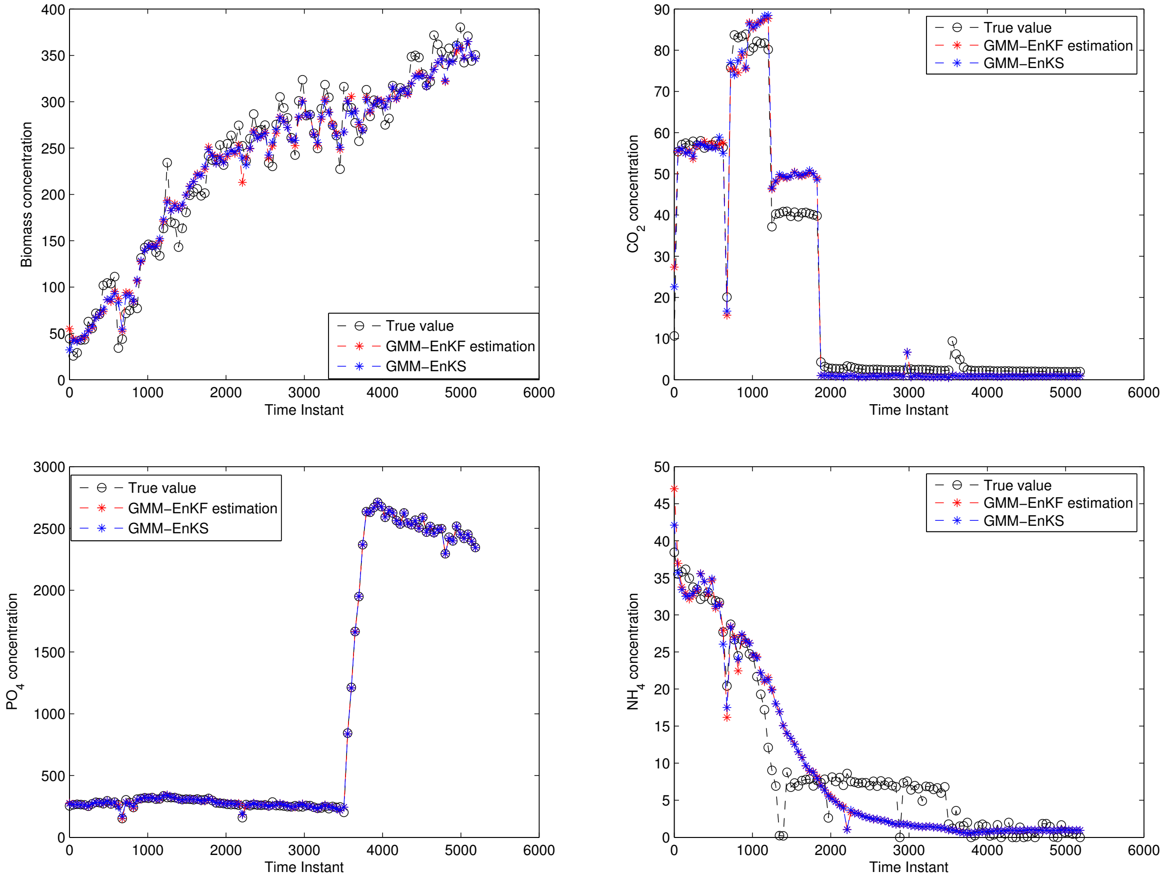


Figure 5.4: Comparison of the GMM-EnKF filtering, GMM-EnKS smoothing results and true values of the four states of the algae process

One key step in our proposed GMM-EnKS is to use the prediction forecast modes to determine the matching forecast mode for the updated mode of last step. In the following,

Table 5.3: The SEE of the four states using GMM-EnKF, GMM-EnKS, EnKF and EnKS

	GMM-EnKF	GMM-EnKS	EnKF	EnKS
X(SSE)	2.9852×10^4	2.8656×10^4	4.4242×10^4	4.1893×10^4
CO_2(SSE)	0.2158×10^4	0.1876×10^4	1.2738×10^4	0.1897×10^4
PO_4(SSE)	0.8740×10^4	0.8649×10^4	1.2738×10^4	0.8612×10^4
NH_4(SSE)	0.1894×10^4	0.1817×10^4	0.1973×10^4	0.1862×10^4

we show the estimation results with and without performing this step. For clarification, the modes of both the forecast and updated step are given order numbers. As there are two modes in this case study, they are numbered as $I = [1, 2]$, where I is the order matrix. By not performing the matching step, it means that the order matrix is always $I = [1, 2]$. Table 5.4 records the smoothing results with and without modifying the order matrix with the proposed algorithm from Eq.(5.37) to Eq.(5.38). Table 5.4 presents the estimation error of using GMM-EnKS with and without the the order matrix I over the first seven steps of smoothing. Note that the estimation error is simply the absolute value of the difference of smoothing result and the true value of each state at every time step. From the table, we can see that the estimation errors for both methods are almost the same because the I matrix is $[1, 2]$, which is the default value of the other case. From the 3rd step, the I matrix is modified, therefore the estimation error is different between the two methods. Although the errors are close to each other, the GMM-EnKS with the I matrix slightly outperforms the other estimator.

5.7 Conclusions

In this chapter, we propose a novel smoother for non-Gaussian and nonlinear systems. The proposed GMM-EnKS extended the EnKS to multimodal PDF by utilizing the same framework as the GMM-EnKF. However, the smoothing involves ensembles of two time steps, which imposes difficulties for the calculation of the cross-covariance in the presence of multimodal PDF. We solve this problem by matching the modes of the GMMs of two steps

Table 5.4: The estimation error with and without order matrix I.

Time step	State estimation error of $[X, CO_2, PO_4, NH_4]$ without order matrix I	State estimation error of $[X, CO_2, PO_4, NH_4]$ with order matrix I	I
1	$[6.5, 11, 0.3, 0.5]$	$[6.5, 11, 0.3, 0.5]$	$[1, 2]$
2	$[5.7, 1, 0.7, 0.4]$	$[5.5, 1, 0.3, 0.4]$	$[1, 2]$
3	$[5.1, 1.1, 0.3, 0.3]$	$[5.1, 1.1, 0.5, 0.3]$	$[2, 1]$
4	$[22.3, 1.1, 4, 0.6]$	$[22.1, 1.1, 5, 0.6]$	$[2, 1]$
5	$[1.3, 1, 0.1, 0.7]$	$[0.5, 1, 0.6, 0.7]$	$[2, 1]$
6	$[4.9, 11, 0.6, 0.9]$	$[4.9, 10, 0.5, 0.9]$	$[2, 1]$
7	$[14.7, 10, 8.4, 0.8]$	$[14.2, 10, 5.9, 0.8]$	$[2, 1]$

with each other and then calculate the covariance matrices. On top of that, we propose to update the probability matrix calculated in the filtering by utilizing the SMC approach. The probability is treated as the weight of each ensemble member, such that the SMC approach can be applied to update it.

Chapter 6

Conclusions and Future Work

6.1 Summary and discussion

The studies described in this dissertation attempt to solve three problems regarding data assimilation techniques for nonlinear and non-Gaussian systems.

- First, a GMM-EnKF filtering approach is developed in attempt to solve the issues arising from the non-Gaussianity of the state PDF. The proposed method retains the philosophy of the EnKF, yet extends the EnKF to scenarios where the prior and posterior distributions are multimodal instead of unimodal by adding an expectation maximization (EM) clustering step before the prediction step. The clustering not only reveals the multimodal structure of the state space, it also provides an analytical expression to describe the GMM used to approximate this multimodality. Most other previous nonlinear filters, except the particle filter, cannot deal with strong non-Gaussianity well. Even the particle filter cannot provide an analytical expression to describe the state PDF. One can only extract the statistical moments for the posterior PDF from its updated particles. The novelty of our version of GMM-EnKF lies in retaining the multimodality of the state PDF throughout the filtering. In addition, another contribution is that we compare it with the EnKF and the PF under multiple scenarios including multimodal and unimodal prior and uncertain parameters and prove that the proposed filter shows more robustness than the PF under plant-model mismatch.

- Second, the GMM-EnKF is extended to the case of smoothing. The proposed smoother, GMM-EnKS, combines our work in the GMM-EnKF with the EnKS framework. Further, we point out the connection between the membership probability in the GMM-EnKS and the weights in the particle smoother and further modify the GMM-EnKS by integrating the weight update in the particle smoother. The GMM-EnKS has shown better performance than the GMM-EnKF. GMM-EnKS with membership weight update shows significant improvement at some time steps, however, it also worsens the filtering results at some other time steps. Moreover, the GMM-EnKS shows significant improvement over the EnKS for multimodal systems.

- Third, another major contribution of the thesis is in state estimation with linear constraints, which includes two aspects-unimodal case and multimodal case. For the unimodal case, novel constrained EKF and EnKF are developed. The constraints are incorporated by designing a convex optimization problem using the Kullback-Leibler (KL) divergence as its objective function, as opposed to the quadratic programming in the recursive nonlinear dynamic data reconciliation (RNDDR) approach. In our simulation, the KL divergence based approach is able to achieve faster convergence rate to the true value with inequality constraints applied. This is because it constrains both the mean and the covariance, which pushes the particles further into the constrained region, closer to the true value. The RNDDR approach, however, tends to retain the particles on the boundary of the constraints in the first few steps of estimation, especially if the measurement noise is large. For the multimodal case, a more complicated optimization problem is designed to solve the constrained state estimation using the previously proposed GMM-EnKF. As there is no analytical expression of the KL divergence between two Gaussian mixtures, we choose to the Goldberger's approximation.

Although Goldberger's approximation is not the most accurate approximation of the KLD of two GMs, we show that it can be formulated into a convex optimization problem which can be implemented online. The constrained GMM-EnKF is able to retain the mixture structure of the distribution and provides an analytical expression for the constrained mixture. In the existing literature, the constrained particle filter is the most feasible solution for non-Gaussian systems, however, it cannot provide an analytical expression of the multimodality as in the constrained GMM-EnKF. Moreover, we show that the performance of the constrained GMM-EnKF is less compromised than the constrained PF when the measurements are unreliable, i.e. the correction of the measurements is weak.

6.2 Future work

- First, in chapter 3 we argue that the reverse KL divergence is more appropriate to serve as the objective function because the constrained mean would lie closer to the unconstrained one than the forward KL divergence. A comparative work using the forward KLD as the objective function needs to be done to further collaborate this theory. One major problem regarding this work is using the forward KLD as the objective function results in a non-convex optimization problem, which might be tricky to solve, especially for the Gaussian mixture case.

- Second, a better approximation than the Goldberger's approximation might be needed for the KL divergence between two Gaussian mixtures. The Goldberger's approximation is based on an assumption that the individual modes of the mixture cannot be overlapping too much. The new approximation approach is expected to relax this limit.

- Third, in the GMM-EnKS, although the update of the membership weights improve the overall performance of the smoothing, it worsens the filtering results in some steps. In our work, the reason behind it is not clarified. In future work, an in-depth analysis is expected to perform to explain the reason.

- Fourth, in the case study of the algae cultivation process, we draw the conclusion that the distribution of the maximum growth rate is multimodal, which means the growth rate might take different values in different growth stages. Further experiments need to be conducted to prove our hypothesis.

Bibliography

[1] D. Alspach and H. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 1972.

[2] B. D. Anderson and J. B. Moore. Optimal filtering. *Englewood Cliffs*, 21:22–95, 1979.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[4] S. Balakrishnapillai Chitralekha. Computational tools for soft sensing and state estimation. 2010.

[5] V. A. Bavdekar, J. Prakash, S. L. Shah, and R. B. Gopaluni. Constrained dual ensemble Kalman filter for state and parameter estimation. In *American Control Conference (ACC), 2013*, pages 3093–3098. IEEE, 2013.

[6] V. A. Bavdekar and S. L. Shah. Computing point estimates from a non-Gaussian posterior distribution using a probabilistic k-means clustering approach. *Journal of Process Control*, 24(2):487–497, 2014.

[7] T. Bengtsson, C. Snyder, and D. Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research: Atmospheres*, 108(D24), 2003.

[8] N. Bergman. Recursive Bayesian estimation. *Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation*, 579:11, 1999.

[9] K. K. Biswas and A. K. Mahalanabis. An approach to fixed-point smoothing problems. *IEEE Transactions on Aerospace and Electronic Systems*, (5):676–682, 1972.

[10] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris. A spatially constrained mixture model for image segmentation. *IEEE transactions on Neural Networks*, 16(2):494–498, 2005.

[11] M. Bocquet and P. Sakov. An iterative ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 140(682):1521–1535, 2014.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[13] S. H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[14] T. Chen, J. Morris, and E. Martin. Particle filters for state and parameter estimation in batch processes. *Journal of Process Control*, 15(6):665–673, 2005.

[15] T. J. Crowley, E. S. Meadows, E. Kostoulas, and F. J. Doyle Iii. Control of particle size distribution described by a population balance model of semibatch emulsion polymerization. *Journal of Process Control*, 10(5):419–432, 2000.

[16] N. De Freitas, C. Andrieu, P. Højen-Sørensen, M. Niranjan, and A. Gee. Sequential Monte Carlo methods for neural networks. In *Sequential Monte Carlo Methods in Practice*, pages 359–379. Springer, 2001.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[18] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

[19] L. Dovera and E. Della Rossa. Multimodal ensemble Kalman filtering using Gaussian mixture models. *Computational Geosciences*, 15(2):307–323, 2011.

[20] F. Doyle, M. Soroush, and C. Cordeiro. Control of product quality in polymerization processes. In *AIChE Symposium Series*, pages 290–306. New York; American Institute of Chemical Engineers; 1998, 2002.

[21] E. B. Dynkin. Markov processes. In *Markov Processes*, pages 77–104. Springer, 1965.

[22] A. A. Emerick and A. C. Reynolds. Ensemble smoother with multiple data assimilation. *Computers & Geosciences*, 55:3–15, 2013.

[23] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[24] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[25] G. Evensen and P. J. Van Leeuwen. An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

[26] J. Flores-Cerrillo and J. F. MacGregor. Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial & Engineering Chemistry Research*, 41(7):1805–1814, 2002.

[27] D. Fraser and J. Potter. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control*, 14(4):387–390, 1969.

[28] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of KL divergence between two gaussian mixtures. In *null*, page 487. IEEE, 2003.

[29] A. Gopalakrishnan, N. S. Kaisare, and S. Narasimhan. Incorporating delayed and infrequent measurements in extended Kalman filter based nonlinear state estimation. *Journal of Process Control*, 21(1):119–129, 2011.

[30] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[31] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[32] H. Greenspan, A. Ruf, and J. Goldberger. Constrained Gaussian mixture model framework for automatic segmentation of mr brain images. *IEEE transactions on medical imaging*, 25(9):1233–1245, 2006.

[33] R. Grønnevik and G. Evensen. Application of ensemble-based techniques in fish stock assessment. *Sarsia*, 86(6):517–526, 2001.

[34] P. Guan, M. Raginsky, and R. M. Willett. Online Markov decision processes with Kullback-Leibler control cost. *IEEE Transactions on Automatic Control*, 59(6):1423–1438, 2014.

[35] N. Gupta and R. Hauser. Kalman filtering with equality and inequality state constraints. *arXiv preprint arXiv:0709.2791*, 2007.

[36] E. L. Haseltine and J. B. Rawlings. Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44(8):2451–2460, 2005.

[37] E. L. Haseltine and J. B. Rawlings. Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Industrial and engineering chemistry research*, 44(8):2451–2460, 2005.

[38] J. R. Hershey and P. A. Olsen. Approximating the Kullback-Leibler divergence between Gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[39] J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.

[40] R. Hostettler. A two filter particle smoother for wiener state-space systems. In *Control Applications (CCA), 2015 IEEE Conference on*, pages 412–417. IEEE, 2015.

[41] X. Hu and L. Xu. Investigation on several model selection criteria for determining the number of clusters. *Neural Information Processing-Letters and Reviews*, 4(1):1–10, 2004.

[42] A. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic, New York, 1972.

[43] A. H. Jazwinski. *Stochastic processes and filtering theory.* Courier Corporation, 2007.

[44] J. H. Jo and S. Bankoff. Digital monitorina and estimation of polymerization reactors. *AIChE Journal*, 22(2):361–369, 1976.

[45] S. J. Julier. The scaled unscented transformation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4555–4559. IEEE, 2002.

[46] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE transactions on communication technology*, 15(1):52–60, 1967.

[47] R. E. Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[48] S. Kasiri, A. Ulrich, and V. Prasad. Kinetic modeling and optimization of carbon dioxide fixation using microalgae cultivated in oil-sands process water. *Chemical Engineering Science*, 137:697–711, 2015.

[49] S. Kasiri, A. Ulrich, and V. Prasad. Optimization of CO2 fixation by Chlorella kessleri cultivated in a closed raceway photo-bioreactor. *Bioresource technology*, 194:144–155, 2015.

[50] C. Keppenne, M. Rienecker, N. Kurkowski, and D. Adamec. Ensemble Kalman filter assimilation of temperature and altimeter data with bias correction and application to seasonal prediction. *Nonlinear Processes in Geophysics*, 12(4):491–503, 2005.

[51] C. Kiparissides. Challenges in particulate polymerization reactor modeling and optimization: A population balance perspective. *Journal of Process Control*, 16(3):205–224, 2006.

[52] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[53] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: If i had a million particles. In *Proceedings of the 23rd International conference on Machine learning*, pages 481–488. ACM, 2006.

[54] S. Kolås, B. A. Foss, and T. Schei. Constrained nonlinear state estimation based on the UKF approach. *Computers & Chemical Engineering*, 33(8):1386–1401, 2009.

[55] J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on signal processing*, 51(10):2602–2612, 2003.

[56] D. J. Kozub and J. F. MacGregor. State estimation for semi-batch polymerization reactors. *Chemical Engineering Science*, 47(5):1047–1062, 1992.

[57] S. V. Kumar, R. H. Reichle, C. D. Peters-Lidard, R. D. Koster, X. Zhan, W. T. Crow, J. B. Eylander, and P. R. Houser. A land surface data assimilation framework using the land information system: Description and applications. *Advances in Water Resources*, 31(11):1419–1432, 2008.

[58] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.

[59] F. L. Lewis and F. Lewis. *Optimal estimation: with an introduction to stochastic control theory*. Wiley New York et al., 1986.

[60] R. Li, V. Prasad, and B. Huang. Gaussian mixture model-based ensemble Kalman filtering for state and parameter estimation for a PMMA process. *Processes*, 4(2):9, 2016.

[61] A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[62] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[63] R. Mandela, V. Kuppuraj, R. Rengaswamy, and S. Narasimhan. Constrained unscented recursive estimator for nonlinear dynamic systems. *Journal of Process Control*, 22(4):718 – 728, 2012.

[64] P. S. Maybeck. *Stochastic models, estimation, and control*, volume 3. Academic press, 1982.

[65] K. McAuley and J. MacGregor. On-line inference of polymer properties in an industrial polyethylene reactor. *AIChE Journal*, 37(6):825–835, 1991.

[66] K. McAuley and J. MacGregor. Nonlinear product property control in industrial gas-phase polyethylene reactors. *AIChE Journal*, 39(5):855–866, 1993.

[67] X. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[68] J. B. Moore. Discrete-time fixed-lag smoothing algorithms. *Automatica*, 9(2):163–173, 1973.

[69] M. Pan and E. F. Wood. Data assimilation for estimating the terrestrial water budget using a constrained ensemble Kalman filter. *Journal of Hydrometeorology*, 7(3):534–547, 2006.

[70] P. Piro, S. Anthoine, E. Debreuve, and M. Barlaud. Image retrieval via Kullback-Leibler divergence of patches of multiscale coefficients in the knn framework. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 230–235. IEEE, 2008.

[71] J. Prakash, S. C. Patwardhan, and S. L. Shah. Constrained nonlinear state estimation using ensemble Kalman filters. *Industrial & Engineering Chemistry Research*, 49(5):2242–2253, 2010.

[72] J. Prakash, S. C. Patwardhan, and S. L. Shah. On the choice of importance distributions for unconstrained and constrained state estimation using particle filter. *Journal of Process Control*, 21(1):3–16, 2011.

[73] V. Prasad, M. Schley, L. P. Russo, and B. W. Bequette. Product property and production rate control of styrene polymerization. *Journal of Process Control*, 12(3):353–372, 2002.

[74] M. L. Psiaki. Backward-smoothing extended Kalman filter. *Journal of guidance control and dynamics*, 28(5):885–894, 2005.

[75] P. N. Raanes. On the ensemble Rauch-Tung-Striebel smoother and its equivalence to the ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 142(696):1259–1264, 2016.

[76] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation: a moving horizon approach. *Automatica*, 37(10):1619 – 1628, 2001.

[77] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

[78] H. E. Rauch, F. Tung, C. T. Striebel, et al. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[79] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review*, 130(1):103–114, 2002.

[80] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8):2209–2224, 1996.

[81] S. Sajjadi and B. Brooks. Unseeded semibatch emulsion polymerization of butyl acrylate: bimodal particle size distribution. *Journal of Polymer Science Part A: Polymer Chemistry*, 38(3):528–545, 2000.

[82] S. Särkkä. Unscented Rauch–Tung–Striebel smoother. *IEEE Transactions on Automatic Control*, 53(3):845–849, 2008.

[83] X. Shao, B. Huang, and J. M. Lee. Constrained Bayesian state estimation–a comparative study and a new particle filter based approach. *Journal of Process Control*, 20(2):143–157, 2010.

[84] A. Shenoy, J. Prakash, V. Prasad, S. Shah, and K. McAuley. Practical issues in state estimation using particle filters: Case studies with polymer reactors. *Journal of Process Control*, 23(2):120–131, 2013.

[85] A. V. Shenoy, V. Prasad, and S. L. Shah. Comparison of unconstrained nonlinear state estimation techniques on a mma polymer reactor. *IFAC Proceedings Volumes*, 43(5):159–164, 2010.

[86] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with em using equivalence constraints. In *Advances in neural information processing systems*, pages 465–472, 2004.

[87] A. Silva-Beard and A. Flores-Tlacuahuac. Effect of process design/operation on the steady-state operability of a methyl methacrylate polymerization reactor. *Industrial & engineering chemistry research*, 38(12):4790–4804, 1999.

[88] D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

[89] J.-A. Skjervheim, G. Evensen, et al. An ensemble smoother for assisted history matching. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2011.

[90] K. W. Smith. Cluster ensemble Kalman filter. *Tellus A*, 59(5):749–757, 2007.

[91] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

[92] G. R. Sriniwas, Y. Arkun, and F. J. Schork. Estimation and control of an α-olefin polymerization reactor. *Journal of Process Control*, 5(5):303–313, 1995.

[93] J. R. Stroud, M. L. Stein, B. M. Lesht, D. J. Schwab, and D. Beletsky. An ensemble Kalman filter and smoother for satellite data assimilation. *Journal of the american statistical association*, 105(491):978–990, 2010.

[94] B. O. Teixeira, L. A. Torres, L. A. Aguirre, and D. S. Bernstein. Unscented filtering for interval-constrained nonlinear systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 5116–5121. IEEE, 2008.

[95] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. In *Advances in neural information processing systems*, pages 599–605, 1999.

[96] S. Ungarala, E. Dolence, and K. Li. Constrained extended Kalman filter for nonlinear state estimation. *IFAC Proceedings Volumes*, 40(5):63 – 68, 2007. 8th IFAC Symposium on Dynamics and Control of Process Systems.

[97] P. Vachhani, S. Narasimhan, and R. Rengaswamy. Robust constrained estimation via unscented transformation. *IFAC Proceedings Volumes*, 37(9):317 – 322, 2004.

[98] P. Vachhani, S. Narasimhan, and R. Rengaswamy. Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation. *Journal of process control*, 16(10):1075–1086, 2006.

[99] P. Vachhani, R. Rengaswamy, V. Gangwal, and S. Narasimhan. Recursive estimation in constrained nonlinear dynamical systems. *AIChE Journal*, 51(3):946–959, 2005.

[100] P. J. Van Leeuwen and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124(12):2898–2913, 1996.

[101] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[102] J. S. Whitaker, G. P. Compo, X. Wei, and T. M. Hamill. Reanalysis without radiosondes using ensemble data assimilation. *Monthly Weather Review*, 132(5):1190–1200, 2004.

[103] J. L. Williams. Gaussian mixture reduction for tracking multiple maneuvering targets in clutter. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT, 2003.

[104] D. Wilson, M. Agarwal, and D. Rippin. Experiences implementing the extended Kalman filter on an industrial batch reactor. *Computers & chemical engineering*, 22(11):1653–1672, 1998.

[105] X. Yang, B. Huang, and V. Prasad. Inequality constrained parameter estimation using filtering approaches. *Chemical Engineering Science*, 106:211–221, 2014.

[106] J. Yin, X. Zhan, Y. Zheng, C. R. Hain, J. Liu, and L. Fang. Optimal ensemble size of ensemble Kalman filter in sequential soil moisture data assimilation. *Geophysical Research Letters*, 42(16):6710–6715, 2015.