BAYESIAN ESTIMATION BY
SEQUENTIAL MONTE CARLO SAMPLING FOR
NONLINEAR DYNAMIC SYSTEMS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Wen-shiang Chen, B.S.

* * * * *

The Ohio State University

2004

Dissertation Committee:

Bhavik R. Bakshi, Co-Adviser
Prem K. Goel, Co-Adviser
Isamu Kusaka
James F. Rathman

Approved by

Co-Adviser

Co-Adviser
Department of Chemical Engineering
Precise estimation of state variables and model parameters is essential for efficient process operation, including model predictive control, abnormal situation management, and decision making under uncertainty. Bayesian formulation of the estimation problem suggests a general solution for all types of systems. Even though the theory of Bayesian estimation of nonlinear dynamic systems has been available for decades, practical implementation has not been feasible due to computational and methodological challenges. Consequently, most existing methods rely on simplifying assumptions to obtain a tractable but approximate solution. For example, extended Kalman filtering (EKF) linearizes the process model and assumes Gaussian prior and noise. Moving horizon based least-squares estimation (MHE) also assumes Gaussian or other fixed-shape prior and noise to obtain a least-squares optimization problem. MHE can impose constraints, but is non-recursive and requires computationally expensive nonlinear or quadratic programming. This dissertation introduces sequential Monte Carlo sampling (SMC) for Bayesian estimation of chemical process systems. This recent approach approximates computationally expensive integration by recursive Monte Carlo sampling, and can obtain accurate estimates of posterior distributions efficiently with minimum assumptions. This approach has not been systematically compared with estimation methods popular for chemical processes, including EKF and MHE. In addition to comparing various estimation methods, this
dissertation also develops a practical framework of SMC for handling process constraints based on an acceptance/rejection algorithm. Furthermore, a novel empirical Bayes approach is developed to deal with practical challenges due to degeneracy and a poor initial guess. The ability of the proposed approach to be more computationally efficient and at least as accurate as MHE is demonstrated via several case studies. A high-dimensional polymerization process is particularly studied to examine the effect of increasing dimensionality on computation load. Empirical results indicate that SMC does not necessarily increase its consumption of CPU cycles dramatically, and may only be slightly dependent on dimensionality. Although this research has only focused on data rectification of nonlinear dynamic systems, the approach is broadly applicable to most process engineering tasks. With increasing computational ability, and theoretical advances, SMC is expected to be an active area of research and application in near future.
Dedicated to my family.
ACKNOWLEDGMENTS

I always feel that I am a lucky person, and am grateful to be able to work with my adviser, Dr. Bhavik R. Bakshi, who inspires me for his foresights in research, passion for innovation, and dedication to excellence. I offer my sincere thanks to him for his support, patience, and encouragements throughout many years of this research.

I am fortunate to work with another profound researcher, Dr. Prem K. Goel, whose insightful knowledge of Bayesian statistics shed a light for me in exploring Bayesian statistics.

Special thanks to Eric L. Haseltine and Dr. James Rawlings at the University of Wisconsin for providing MHE code and assistance in running the program.

I am grateful to Dr. Sridhar Ungarala for initializing the studies of Bayesian statistics in his post-doctoral research under Dr. Bhavik R. Bakshi.

I thank Dr. Liang-Shih Fan for his help in my admission to the Ohio State University, and financial support of my early graduate life from 1997 to 1999.

Financial support from the National Science Foundation (CTS-9733627 and CTS-0321911) is also gratefully acknowledged.
VITA

February 10, 1971  ......................... Born - Taipei, Taiwan

1993  ........................................B.S. Chemical Engineering

1996-present  .................................Graduate Research Associate,
The Ohio State University.

FIELDS OF STUDY

Major Field: Chemical Engineering

Studies in:

- Estimation by Bayesian Statistics
- Multi-scale analysis: wavelet transformation
- Nonlinear dynamic analysis: chaos theory
# TABLE OF CONTENTS

Abstract ................................................................. ii

Dedication ............................................................... iv

Acknowledgments ....................................................... v

Vita ................................................................. vi

List of Tables ......................................................... x

List of Figures ......................................................... xii

Chapters:

1. Introduction ....................................................... 1
   1.1 Motivation ...................................................... 1
   1.2 Results and Contributions .................................... 4
   1.3 Outline ......................................................... 6

2. Literature Review ................................................ 8
   2.1 Estimation Problems ........................................... 8
   2.2 Bayesian Approach to Dynamic Data Rectification .......... 10
   2.3 Bayesian view of existing methods .............................. 13
      2.3.1 Extended Kalman Filtering ................................. 14
      2.3.2 Moving Horizon Based Least-Squares Estimation ....... 15
6.2 Illustrations and Case Studies .................................. 82
   6.2.1 Process Model and Simulation Conditions ............... 82
   6.2.2 Cases Studied ............................................. 87
   6.2.3 Effect of Dimension on Computation Load ............... 95
6.3 Summary ....................................................... 96
6.4 Notation ....................................................... 96

7. Conclusions and Future Work .................................. 98
   7.1 Conclusions .................................................. 98
   7.2 Future Work ................................................ 101

Appendices:

A. Source Code .................................................... 104

B. Derivation of Existing Methods via Bayesian Estimation ...... 110
   B.1 Derivation of Kalman Filtering via Bayesian Estimation ...... 110
   B.2 Derivation of Moving Horizon Estimation via Bayesian Estimation 112

Bibliography ....................................................... 114
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Assumptions required for the convergence of MHE.</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Effect of the support of importance functions on convergence rate.</td>
<td>26</td>
</tr>
<tr>
<td>3.2 General algorithm of SMC.</td>
<td>28</td>
</tr>
<tr>
<td>3.3 Weighted bootstrap filtering algorithm for SMC.</td>
<td>40</td>
</tr>
<tr>
<td>3.4 Hybrid gradient descent/sampling importance resampling algorithm for SMC.</td>
<td>42</td>
</tr>
<tr>
<td>3.5 Empirical Bayes sampling importance resampling algorithm for SMC.</td>
<td>43</td>
</tr>
<tr>
<td>4.1 Mean-squares error and CPU load of linear Gaussian system.</td>
<td>46</td>
</tr>
<tr>
<td>4.2 Mean-squares error and CPU load of nonlinear dynamic system.</td>
<td>48</td>
</tr>
<tr>
<td>4.3 First operating condition of unconstrained CSTR.</td>
<td>50</td>
</tr>
<tr>
<td>4.4 Mean-squares error and CPU load of unconstrained CSTR for first operating condition with perfect initial guess.</td>
<td>51</td>
</tr>
<tr>
<td>4.5 Mean-squares error and CPU load of unconstrained CSTR for first operating condition with poor initial guess.</td>
<td>53</td>
</tr>
<tr>
<td>4.6 Second operating condition of unconstrained CSTR.</td>
<td>54</td>
</tr>
<tr>
<td>4.7 Mean-squares error and CPU load of unconstrained CSTR for second operating condition.</td>
<td>57</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.1 Algorithm of SMC with enforcement of constraints.</td>
<td>63</td>
</tr>
<tr>
<td>5.2 Mean-squares error and CPU load of constrained linear system.</td>
<td>67</td>
</tr>
<tr>
<td>5.3 Mean-squares error and CPU load of constrained CSTR with Gaussian noise.</td>
<td>69</td>
</tr>
<tr>
<td>5.4 Parameters of double-half normal distributions of non-Gaussian noise.</td>
<td>72</td>
</tr>
<tr>
<td>5.5 Mean-squares error and CPU load of constrained CSTR with non-Gaussian noise.</td>
<td>74</td>
</tr>
<tr>
<td>6.1 Simulation parameters of polymerization process.</td>
<td>85</td>
</tr>
<tr>
<td>6.2 Operating conditions and simulation parameters of polymerization process.</td>
<td>85</td>
</tr>
<tr>
<td>6.3 Operating conditions and simulation parameters of polymerization process.</td>
<td>86</td>
</tr>
<tr>
<td>6.4 Mean-squares error and CPU load of 2-D polymerization process.</td>
<td>89</td>
</tr>
<tr>
<td>6.5 Mean-squares error and CPU load of 5-D polymerization process.</td>
<td>91</td>
</tr>
<tr>
<td>6.6 Mean-squares error and CPU load of 8-D polymerization process.</td>
<td>94</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>52</td>
</tr>
<tr>
<td>4.5</td>
<td>52</td>
</tr>
<tr>
<td>4.6</td>
<td>54</td>
</tr>
</tbody>
</table>
4.7 Estimation performance of unconstrained CSTR for first operating condition with poor initial guess. (a) Overall MSE. (b) MSE after initial dynamics. (c) CPU load. .............................................. 55

4.8 Evolution of the posterior of the concentration of unconstrained CSTR for the second operating condition. ........................... 56

4.9 Estimation performance of unconstrained CSTR for second operating condition. ......................................................... 57

5.1 Evolution of the posterior of the concentration of constrained CSTR. ................................................................. 60

5.2 Histogram of non-Gaussian noise (left). Normplot of non-Gaussian noise (right). ................................................................. 66

5.3 Estimation performance of constrained linear system. ....................... 68

5.4 Rectification result of $x_1$ of constrained linear system. ...................... 68

5.5 Estimation performance of constrained CSTR with Gaussian noise. . 70

5.6 Rectification result of the concentration of constrained CSTR with Gaussian noise. ......................................................... 71

5.7 Rectification result of the temperature of constrained CSTR with Gaussian noise. ................................................................. 71

5.8 Approximation of a non-Gaussian distribution, $p(\nu_1)$, by double-half Gaussian approximation (left) and Monte Carlo approximation. .... 72

5.9 Estimation performance of constrained CSTR with non-Gaussian noise. 74

5.10 Rectification result of the concentration of constrained CSTR with non-Gaussian noise. ......................................................... 75

5.11 Rectification result of the temperature of constrained CSTR with non-Gaussian noise. ......................................................... 75
5.12 Illustration of the broaden variance of constrained CSTR with non-Gaussian noise. Left hand side one has a narrower marginal confidence interval on concentration than the right hand side one.  

6.1 Accurate estimation by Monte Carlo sampling may only need limited number of samples that well represent the key features of the posterior (as illustrated in colored boxes in (a) and (b)), thus may not need a set of samples which cover the whole solution space (as illustrated in (c) and (d)).  

6.2 Rectification result of the temperature in 2-D polymerization process.  

6.3 Estimation performance of 2-D polymerization process.  

6.4 Rectification result of the temperature of 5-D polymerization process.  

6.5 Estimation performance of 5-D polymerization process.  

6.6 Rectification result of the temperature of 8-D polymerization process.  

6.7 Estimation performance of 8-D polymerization process.  

6.8 Increment of CPU load of MHE and SMC according to dimensionality.  

A.1 Flow chart of SMC code.
CHAPTER 1

INTRODUCTION

1.1 Motivation

Modern industrial enterprises have invested significant resources for collecting and distributing data, with the expectation that it will enhance profitability via better decision making. However, realizing this potential requires techniques for decision making via extracting useful information from measured data, fundamental and empirical models, constraints and experience based knowledge. This need is common across most process engineering tasks including, feed-back control, process monitoring, fault detection and diagnosis, planning and scheduling, process design, and scale-up. Due to the importance of these tasks, many methods have been developed under the names of data rectification, data reconciliation, and state and parameter estimation [23, 34].

An ideal technique for tackling such complicated problems should have the following features. First, it should handle all kinds of data, models and information. This includes missing and multi-rate data, fundamental and empirical models, nonlinear dynamic systems, and experience-based knowledge. This approach should also be
able to extract maximum information via a scientifically rigorous approach. Furthermore, it should permit efficient computation for on-line or off-line use. Finally, this ideal method should provide information about uncertainty in the decision making.

Despite significant efforts, these goals have been quite challenging, and most methods rely on a variety of simplifying assumptions. Most of existing approaches developed are known to make invalid, but convenient assumptions. For example, extended Kalman filtering (EKF) relies on Gaussian approximation and local linearization to find a computationally efficient solution. Moving horizon based least-squares estimation (MHE), also relies on Gaussian approximation so that a least-squares formulation may be found. These assumptions may work fine with linear dynamic systems without constraints, but can be easily violated in nonlinear dynamic systems. Figure 1.1 illustrates the dynamic evolution of the posterior over time for a popular continuously stirred tank reactor (CSTR) case study [19, 24, 16, 34]. These distributions are obtained by the Monte Carlo approach described in Section 3.2 with a Gaussian initial guess. The multi-modal, skewed and time-varying nature of these distributions indicates that approximating them by Gaussian or other fixed-shape distributions can be grossly incorrect. These approximate distributions can be even worse in the presence of constraints since the probability of some variables in regions where the constraints are violated must be zero [34, 29, 33]. Nevertheless, these assumptions are popular since they permit existing methods to solve a convenient problem instead of the actual estimation problem. These shortcomings of existing methods are well-known and have been widely recognized [34, 29, 33].

Bayesian formulation of the estimation problem suggests a general strategy for all types of systems. Although the underlying theory and equations for the proposed
Figure 1.1: Evolution of the posterior of the concentration of unconstrained CSTR.

approach have been available for at least four decades [18], and many attempts have been made for using this formulation in developing practical methods for on-line estimation. However, direct integration of these equations continues to be prohibitively expensive. Consequently, only methods based on simplifying assumptions have been practical for estimation of nonlinear dynamic systems [34, 29, 33]. Recent theoretical advances coupled with fast computation are fueling a revival in Bayesian statistics [25, 37]. Among available techniques, an approach based on a rigorous Bayesian formulation that uses sequential Monte Carlo (SMC) sampling to propagate all information recursively has shown promising results in providing accurate and efficient estimates [14, 9, 11]. However, existing approaches of SMC have yet to address various issues of engineering practices. For example, process constraints commonly
exist in engineering practices, like constraints based on mass or energy balance, or constraints that dictate physical variables to be non-negative. So far, there is no practical algorithm for SMC devoted to the handling of process constraints. In addition, most published work is based on the results of relatively low-dimensional problems. Even though theoretical studies of SMC have indicated that SMC may be free of the “curse of dimensionality” [7], there is no clear indication how SMC may perform empirically in high-dimensional cases. This research intends to introduce SMC methods for estimation of process systems engineering. In addition to direct application of the available algorithm, a genuine formulation is developed to facilitate the use of process constraints. Finally, an empirical exploration of the effect of dimensionality on the computation load of SMC is proceeded.

1.2 Results and Contributions

This dissertation introduces a novel approach for estimation in process system engineering. The fundamental theory of this research, a recursive formulation of Bayesian estimation, has been proposed at least for decades, but was not considered practical until recently. This perception of infeasibility of Bayesian estimation arose due to the prohibitively expensive computation load by direct integration. However, advances in Bayesian statistics, and affordable powerful computers have made Bayesian methods promising for practical usage in various areas of research. In this dissertation, claims of SMC to provide more accurate estimation and excitingly less computation load have been shown by various cases studies in the area of data rectification, including linear and nonlinear, Gaussian and non-Gaussian, constrained and non-constrained systems. Comparison of various estimation methods, including
EKF, MHE and SMC, are based on the average mean-squares error (MSE) and CPU load with 100 realizations. No such comparison has been provided in the past. As expected, SMC shows significant improvement on estimation accurate especially when the underlying distributions exhibit non-Gaussian features. SMC exhibits an unexpected advantage of consuming much less computation power over MHE even when all MHE’s assumptions are satisfied and a customized optimization algorithm, successive quadratic programming (SQP), is used. These results indicate that Gaussian approximation does not necessarily provide computation benefit.

Bayesian estimation is expected to provide more valuable information of the posterior than existing methods which usually merely provide point estimates as the estimation result. In general, point estimates are poor representation of multi-modal distributions, and nonlinear dynamic systems actually have been shown to exhibit such features (Figure 1.1). Discussion of the alternative methods of comparing point estimates is given in Section 4.1. In this dissertation, boundary points of the high probability density of the posterior are provided to illustrate the advantage of having the whole posterior as the estimate over approaches that merely provide point estimates as the estimation result.

A genuine approach which permits the use of constraints of SMC has also been developed in this research. Enforcement of constraints is implemented with an acceptance/rejection algorithm that keeps qualified samples, while rejects the rest. This implementation is simple and straightforward, thus minor increment in CPU load has been observed (comparing cases in Section 4.2.3 and 5.2.2). EKF is disadvantaged by not enforcing constraints, and does result in inferior estimation accuracy as shown in Section 5.2. MHE has more accurate estimates than EKF, but consistently to be the
most expensive approach in computation. SMC outperforms all methods in accuracy especially when the underlying distributions are non-Gaussian.

Feasibility of SMC in large-scale problems has also be studied in this dissertation. SMC has been indicated to be free of the “curse of dimensionality” based on limited available literatures [7]. However, the performance of SMC for large-scale problems still needs to be investigated further. A detail examination of SMC in high-dimensional problems is expected to provide insights to answer the practical values of SMC in real life large-scale problems. This research addresses this issue by studying an eight dimensional polymerization process, and examine the effect of increasing dimensionality on computing load. Results show that the CPU load does increase to compensate the increased complexity, but in a rate implies that SMC may not explode in computation even when in high-dimensional systems. Even if SMC might require much more samples for accurate estimation, it is highly likely that SMC can work with the help of parallel computation.

Furthermore, this dissertation proposes a novel method for dealing with the practical challenges in handling degeneracy and poor initial guess. This approach, called Empirical Bayes Sampling Importance Resampling (EBSIR), provides better accuracy with more convenient tuning parameters than existing SMC methods. Potentials of Bayesian estimation in engineering practice is also explored in this paper.

1.3 Outline

The rest of the dissertation is organized as follows. A general formulation of the estimation problem is introduced in Chapter 2. In the same chapter, challenges faced by a rigorous Bayesian formulation are discussed, and how existing approaches
tackled these issues. The proposed approach, SMC, is introduced in Chapter 3, with background information of Monte Carlo sampling. Practical issues of SMC are also discussed in the same chapter, followed by a short survey of the convergence of SMC. Several practical solutions, including the proposed EBSIR, are explained in the last section of Chapter 3.

Three areas of applications are covered in this dissertation, including the application to unconstrained systems in Chapter 4, the application to constrained systems in Chapter 5, and the study of SMC and the “curse of dimensionality” in Chapter 6. In Chapter 4, SMC is first verified by the estimation result of Kalman filtering, the optimal estimator in a linear Gaussian system. Then a benchmark nonlinear dynamic systems which exhibits significant non-Gaussian features is studied. Then the application of SMC to a well studied chemical engineering problem, a continuously stirred tank reactor (CSTR) is studied. Two cases are studied in the constrained cases of Chapter 5, including a linear dynamic system and the same CSTR problem in the unconstrained chapter, but with constraints in this chapter. The CSTR problem is further examined with the existence of non-Gaussian measurement noise to demonstrate the flexibility of application of SMC. Finally, the increment in computation load along with the increment of dimensionality is studied in Chapter 7 with a eight-dimensional polymerization process.
CHAPTER 2

LITERATURE REVIEW

2.1 Estimation Problems

In general, the goal of estimation may be expressed as follows. Given measurements \( y_{1:k} = \{y_1, y_2, \ldots, y_k\} \), process model and initial guess \( p(x_0) \), determine the current state, \( x_k \). Process models may be expressed as follows,

\[
x_k = f_{k-1}(x_{k-1}, \omega_{k-1}) , \quad (2.1)
\]

\[
y_k = h_k(x_k, \nu_k) , \quad (2.2)
\]

\[
g_1(x_k, y_k, x_{k-1}, \omega_{k-1}, \nu_k) = 0 , \quad (2.3)
\]

\[
g_2(x_k, y_k, x_{k-1}, \omega_{k-1}, \nu_k) \leq 0 , \quad (2.4)
\]

Here, \( x_k \in \mathbb{R}^{n_x} \) is the state vector and \( f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\omega} \rightarrow \mathbb{R}^{n_x} \) is the system equation. Measurements, \( y_k \in \mathbb{R}^{n_y} \), are related to the state vector through the measurement equation, \( h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\nu} \rightarrow \mathbb{R}^{n_y} \). System noise, \( \omega_{k-1} \in \mathbb{R}^{n_\omega} \), represents disturbances in the system, and measurement noise, \( \nu_k \in \mathbb{R}^{n_\nu} \) captures the inaccuracy in measuring systems. Process constraints are captured by either equality constraints, \( g_1(\cdot) \), or inequality constraints, \( g_2(\cdot) \).
Most previous research has focused on rectification of data from linear steady state and linear unconstrained dynamic systems [8, 23, 27, 35]. Significant efforts have also focused on methods for rectification and estimation in nonlinear dynamic systems, with and without constraints [19, 42, 24, 34, 30]. However, all existing methods rely on assumptions about the nature of the model or the probability distributions of the underlying variables to obtain a tractable optimization problem. A popular assumption is that the distribution of the variables to be estimated is Gaussian or of a fixed, time-invariant shape. This assumption permits the formulation of nonlinear dynamic data rectification (NDDR) as a convenient least-squares problem. The failure of this assumption is depicted in Figure 1.1, which shows the actual posterior distribution over time for a popular continuously stirred tank reactor (CSTR) case study [19, 24, 16, 34]. These distributions are obtained by the Monte Carlo approach described in Section 3.2 with a Gaussian initial guess. The multi-modal, skewed and time-varying nature of these distributions indicates that approximating them by Gaussian or other fixed-shape distributions can be grossly incorrect. These approximate distributions can be even worse in the presence of constraints since the probability of some variables in regions where the constraints are violated must be zero [34, 29, 33]. Nevertheless, these assumptions are popular since they permit existing methods to solve a convenient problem instead of the actual estimation problem. These shortcomings of existing methods are well-known and have been widely recognized [34, 29, 33].

This dissertation develops a fundamentally different and strictly Bayesian approach to rectification or estimation of unknown quantities from nonlinear dynamic systems. The underlying theory and equations for the proposed approach have been
available for at least four decades [18], and many attempts have been made for using this formulation in developing practical methods for on-line estimation. However, direct integration of these equations continues to be prohibitively expensive. Consequently, only methods based on simplifying assumptions have been practical for estimation of nonlinear dynamic systems [34, 29, 33]. Recent theoretical advances coupled with fast computation are fueling a revival in Bayesian statistics [25, 37]. These developments provide the foundation for the work described in this thesis.

The rest of the chapter is organized as follows, first, a brief review of Bayesian statistics is provided. Then connection between Bayesian methods and existing approaches is discussed.

2.2 Bayesian Approach to Dynamic Data Rectification

Bayesian estimation provides a rigorous approach for estimating the probability distribution of unknown variables by utilizing all the available knowledge, data and information about the system. It considers all the variables to be stochastic and determines the distribution of the variables to be estimated, \( x \), given the measurements, \( y \). Using the generic symbol \( p \) to denote any probability density function, the conditional density function \( x \) given \( y \) is obtained via Bayes rule as,

\[
p(x|y) = \frac{p(y|x) \ p(x)}{p(y)} , \quad \propto \ p(y|x) \ p(x) . \tag{2.5}
\]

Information contained in the current measurement is represented by the likelihood, \( p(y|x) \), while prior knowledge about the unknown variables is represented by \( p(x) \). The denominator, \( p(y) \), is the evidence provided by the measurements, and is a normalizing constant. Thus, Equation (2.5) combines prior and current knowledge to
obtain the *a posteriori* information of the system. Bayesian estimation can handle all kinds of distributions in prior, likelihood and posterior.

For dynamic systems, a recursive formulation of Bayesian estimation may be represented as follows [18].

\[
p(x_k|y_{1:k}) = \frac{p(y_k|x_k) \ p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})},
\]

\[
\propto p(y_k|x_k) \ p(x_k|y_{1:k-1}), \quad (2.6)
\]

where previous knowledge of the system, \(p(x_k|y_{1:k-1})\), is combined with the most current information of the system, \(p(y_k|x_k)\), to find the posterior, \(p(x_k|y_{1:k})\). Using the relationship of \(p(a) = \int p(a|b)p(b)db\), each term in Equation (2.6) may be obtained as follows. For the prior, \(p(x_k|y_{1:k-1})\), may be found as following,

\[
p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}, y_{1:k-1}) \ p(x_{k-1}|y_{1:k-1}) \ dx_{k-1}, \quad (2.7)
\]

\[
= \int p(x_k|x_{k-1}) \ p(x_{k-1}|y_{1:k-1}) \ dx_{k-1}, \quad (2.8)
\]

where \(p(x_{k-1}|y_{1:k-1})\) is the posterior at time \(k-1\). In Equation (2.7), \(p(x_k|x_{k-1}, y_{1:k-1})\) reduces to \(p(x_k|x_{k-1})\) because the system model (Equation (2.1)) is a Markov process. Further simplification of \(p(x_k|x_{k-1})\) is provided as follows.

\[
p(x_k|x_{k-1}) = \int p(x_k|x_{k-1}, \omega_{k-1}) \ p(\omega_{k-1}|x_{k-1}) \ d\omega_{k-1},
\]

\[
= \int \delta(x_k - f_{k-1}(x_{k-1}, \omega_{k-1})) \ p(\omega_{k-1}) \ d\omega_{k-1}, \quad (2.9)
\]

where \(\delta(\cdot)\) is the Dirac delta function which arises because \(x_k\) can be exactly determined when \(x_{k-1}\) and \(\omega_{k-1}\) are known. Furthermore, \(p(\omega_{k-1}|x_{k-1})\) can reduce to \(p(\omega_{k-1})\) with the assumption that the noise is independent of the current state. Likewise, \(p(y_k|x_k)\) may be found as,

\[
p(y_k|x_k) = \int p(y_k|x_k, \nu_k) \ p(\nu_k|x_k) \ d\nu_k,
\]
Figure 2.1: Algorithm of recursive Bayesian estimation.

\[
\begin{align*}
\int \delta(y_k - h_k(x_k, \nu_k)) \, p(\nu_k) \, d\nu_k.
\end{align*}
\] (2.10)

The recursive algorithm of Bayesian estimation may be visualized as in Figure 2.1. Information in previous measurements up to time \(k-1\) is captured by \(p(x_{k-1}|y_{1:k-1})\). Prediction of distribution of the current state is implemented by utilizing Equations (2.8) and (2.9). Information in current measurement is represented as the likelihood function based on Equation (2.10). The posterior can then be found by combining previous and current information by Equation (2.6).

After the posterior is available, the optimal point estimate, \(\hat{x}_k\), corresponding to the loss function, \(\mathcal{L}(x_k, \hat{x}_k)\), may be obtained by optimizing the objective function [20],

\[
\min_{\hat{x}_k} E[\mathcal{L}(x_k, \hat{x}_k)] = \min_{\hat{x}_k} \int \mathcal{L}(x_k, \hat{x}_k) \, p(x_k|y_{1:k}) \, dx_k.
\]

Bayesian estimation can use any loss function without changing its basic formulation and can readily provide error bounds. This is an advantage over many existing approaches, including MHE, that focus on point estimation. Various kinds of loss
functions exist, providing popular choice of optimal estimates, namely the mean, median or mode of a distribution [20, 18]. However, such point estimates are often inadequate for multi-modal, or non-Gaussian posterior distributions.

In general, there is no closed-form solution for Equations (2.10) to (2.9). Direct integration is computationally expensive and may not be practical for high-dimensional systems. Most estimation methods address these computational challenges by making simplifying assumptions about the nature of the model and/or posterior distributions at the cost of accuracy and computational efficiency. However, recent theoretical advances coupled with fast computation are providing the foundation of building a feasible Bayesian approach even for large scale systems. This computationally efficient algorithm is based on sequential Monte Carlo sampling and will be discussed in Section 3.2.3.

2.3 Bayesian view of existing methods

This section provides a Bayesian view of existing methods by focusing on the approach for solving the equations in Section 2.2. Each method is interpreted as a variation of Bayesian estimation depending on approximations for making the solution more convenient. One common assumption underlying existing methods is that these methods tend to assume a Gaussian or some other specified shape for the prior which is time-invariant. Gaussian distributions are convenient approximations, since closed-form solutions may be obtained for Equation (2.8) to (2.10) and only two parameters, mean and variance, are required to describe the entire distribution. Although the assumption that variables are distributed as Gaussian at all times is
often acceptable in linear systems, it can be easily violated in nonlinear dynamic systems as shown in Figure 1.1. The assumption of Gaussian prior can become highly inaccurate when process constraints are enforced and results in truncated distributions [33, 30]. Nevertheless, approaches based on this assumption are popular since it may lead to computationally efficient methods. Two popular variations of Gaussian approximation are EKF and MHE.

### 2.3.1 Extended Kalman Filtering

Extended Kalman Filtering (EKF) is an extension of Kalman filtering (KF) and may be better explained via a brief review of KF. KF is the optimal estimator for unconstrained linear dynamic systems with Gaussian and additive independent and identically distributed (iid) noise. It is popular due to its optimality and availability of a closed-form solution that makes estimation extremely efficient [20, 26].

For linear dynamic systems with the following process model,

\begin{align}
  x_k &= F_{k-1} x_{k-1} + \omega_{k-1}, \\
  y_k &= H_k x_k + \nu_k,
\end{align}

the posterior may be found as,

\begin{equation}
\log[p(x_k|y_{1:k})] \propto -\{(y_k - H_k x_k)^T Q_{\nu_k}^{-1} (y_k - H_k x_k) + (x_k - F_{k-1} x_{k-1})^T Q_{\omega_k-1}^{-1} (x_k - F_{k-1} x_{k-1}) \\
+ (x_{k-1} - \mu_{x_{k-1}})^T Q_{x_{k-1}}^{-1} (x_{k-1} - \mu_{x_{k-1}})\},
\end{equation}

where $Q_{\nu_k}$ is the covariance of the measurement noise, $\nu_k$, and $Q_{\omega_k-1}$ is the covariance of the system noise, $\omega_{k-1}$. For simplicity, error covariance is assumed to be available for all methods in this dissertation. For linear dynamic systems with Gaussian prior
and additive Gaussian noise, the posterior is also a Gaussian distribution [34]. The prior of next time step then also becomes a Gaussian distribution. This constitutes a recursive formulation for estimation. Equation (2.13) demonstrates that maximizing the posterior is equivalent to minimizing the quadratic terms. KF can be derived by having the same least-squares formulation as the objective function [20].

EKF extends the application of KF to estimation of nonlinear dynamic systems by linearizing the nonlinear process model at each time step so that the same solution strategy of KF can be applied. In doing so, EKF inherits all the benefits of KF, including efficient computation, and all the assumptions, including Gaussian prior and additive noise. EKF then solves a simplified problem in addition to its limitation on problems with additive Gaussian noise.

EKF is favored for its simplicity and efficiency, but may diverge from the true state and need not satisfy process constraints. The divergence issue may be alleviated by methods such as retaining higher order terms of Taylor’s expansion during linearization. However, the optimization solution may not have a closed-form solution anymore and determining the best model order may not be a trivial task. Further discussion of EKF can also be found in Jazwinski [20] and Maybeck [26].

2.3.2 Moving Horizon Based Least-Squares Estimation

Moving Horizon Based Least-Squares Estimation (MHE) is tailored to satisfy constraints in estimation of linear and nonlinear dynamic systems. It also relies on the assumption of Gaussian prior and noise to obtain a least-squares estimation (LSE) problem [34]. LSE may be convenient for finding point estimates, like the mode of the posterior, but usually lacks the ability to find all vital information of the posterior for
propagation, including the covariance, skewness, and number of modes. This incomplete estimation of the posterior usually implies that there is no practical algorithm for MHE to fully utilize the current estimate for future use. Therefore MHE relies on sub-optimal methods to fill-in the missing information, including an external estimator, like EKF, or a smooth estimator based on local linearization [41]. This tends to result in inferior prior, and demands an algorithm with a horizon window to improve the estimation accuracy. This requirement of a horizon window prevents MHE from being a full recursive formulation, like KF. In addition, a closed-form solution is not available anymore. Instead, MHE needs to solve a constrained optimization problem over each moving window. These make MHE a computationally demanding algorithm even when its assumptions are satisfied and a customized optimization algorithm like successive quadratic programming (SQP) is used [34]. MHE is favored for constrained NDDR problems since it is usually more stable than EKF and can handle constraints. However, it needs careful selection of the window size to have accurate and efficient estimation. In addition, use of constrained nonlinear optimization solvers requires careful problem formulation and initialization.

Like EKF, MHE simplifies Equations (2.9) and (2.10) with the assumption of additive iid Gaussian noise. For systems with the following process model, and no additional constraints,

\[
x_k = f^*_k(x_{k-1}) + \omega_{k-1},
\]

\[
y_k = h^*_k(x_k) + \nu_{k-1},
\]

the posterior may be written as,
\[
\log[p(x_{k-m+1:k}|y_{1:k})] \propto \\
- \left\{ \sum_{i=1}^{m} (y_{k-m+i} - h^*_k-m+i(x_{k-m+i}))^T Q_{x_{k-m+i}}^{-1} (y_{k-m+i} - h^*_k-m+i(x_{k-m+i})) \right\} \\
+ \left\{ \sum_{i=1}^{m-1} (x_{k-m+i+1} - f^*_k-m+i(x_{k-m+i}))^T Q_{x_{k-m+i}}^{-1} (x_{k-m+i+1} - f^*_k-m+i(x_{k-m+i})) \right\} \\
+ (x_{k-m+1} - f^*_k-m(\mu_{x_{k-m}}))^T Q_{x_{k-m}}^{-1} (x_{k-m+1} - f^*_k-m(\mu_{x_{k-m}})) \right\} , \tag{2.14}
\]

where the prediction of state \( k - m + 1 \) is assumed to be Gaussian with mean \( f^*_k-m(\mu_{x_{k-m}}) \) and variance \( Q_{x_{k-m}}^{-1} \). The maximum a posteriori (MAP) estimation of Equation (2.14) results in the same objective function as MHE [34]. In general, the optimization problem of MHE based on Equation (2.14) lacks a closed-form solution. Propagation of prior poses another challenge in accurate estimation by MHE, and EKF is usually used for this task.

Extension application of MHE into situations when non-Gaussian distributions occur has been developed. As illustrated in Figure 1.1, non-Gaussian prior is not uncommon in nonlinear dynamic systems. In addition, with the presence of constraints, MHE implicitly uses a truncated Gaussian prior. One suggestion is to approximate non-Gaussian distributions with mixtures of Gaussian distributions [30], in which case some simplifications for optimization may be achieved for least-squares type criterion function. However, Gaussian sum method still assumes a pre-determined shape of distributions, thus may not be practical for approximating the dynamically changing prior. In addition, increased accuracy may require a large number of Gaussian distributions which may make this Gaussian sum method infeasible especially for large scale problems.
Survey of the Convergence of MHE

Asymptotical convergence of the estimate of MHE toward the true state is guaranteed, and the estimate error is bounded under restrictive assumptions [31]. Before the actual introduction of the required assumptions, various terms are defined first.

First, the goal of MHE may be formulated as the following equation [31],

$$\Phi_k = \min_{x_0, \{\varphi_j\}} \left\{ \sum_{j=0}^{k-1} L_j(\varphi_j, \nu_j) + \Gamma(x_0) \right\} ,$$

(2.15)

where $L_j(\cdot)$ is the stage cost function at time $j$, and accounts for the uncertainty due to the existence of noise. Function $\Gamma(\cdot)$ is the initial penalty which captures the deviation of the initial guess to the true initial state. This formulation is slightly more general than the earlier suggestion of MHE, which already imposes a quadratic formulation on $L_j(\cdot)$ [34]. Compared to the Bayesian formulation of Equations (2.8) to (2.10), $\Gamma(\cdot)$ may be represented as the prior, $p(x_0)$, and $L_k(\cdot)$ may be captured by $p(x_k|x_{k-1})$ and $p(y_k|x_k)$.

The problem formulated in Equation (2.15) grows at least linearly with the arrival of new measurements [34]. In order to limit the size of the problem, thus feasible for online applications, a moving horizon formulation is required and may be formulated as follows.

$$\Phi_k = \min_{z, \{\varphi_j\} k-N_{mhe}} \left\{ \sum_{j=k-N_{mhe}}^{k-1} L_j(\varphi_j, \nu_j) + Z_{k-N_{mhe}}(z) \right\} ,$$

where $N_{mhe}$ is the horizon width of MHE, and $Z_{k-N_{mhe}}(\cdot)$ is the *arrival cost*, which contains the information of $\{\sum_{j=0}^{k-N_{mhe}-1} L_j(\varphi_j, \nu_j) + \Gamma(x_0)\}$. Arrival cost $Z_k$ may be considered as $p(x_k|y_{1:k-1})$, conditioned on $x_{k-mhe}|x_0 = z$. However, in general, there is no closed-form solution of $Z_k$, and an approximation is needed. The actual problem
1: \( f_k(\cdot) \) and \( h_k(\cdot) \) are twice differentiable,

2: \( L_k(\cdot) \) and \( \Gamma_k(\cdot) \) are positive definite quadratic functions,

3: \( 0 \leq \hat{Z}_k(z) - \Phi_k \leq \gamma(||z - \hat{x}_{k,mhe}||) \),

4: \( \hat{Z}_k(z) \leq \min_{\omega_j, j = k - N_{mhe}} \left\{ \sum_{j = k - N_{mhe}}^{k-1} L_j(\omega_j, \nu_j) + \hat{Z}_{k-N_{mhe}}(z) \right\} \)

s.t. \( x_k | (x_{k-N_{mhe}} = m) = z \),

Table 2.1: Assumptions required for the convergence of MHE.

solved by MHE is then expressed as the following equation,

\[
\hat{\Phi}_k = \min_{z, \omega_j, j = k-N_{mhe}} \left\{ \sum_{j = k-N_{mhe}}^{k-1} L_j(\omega_j, \nu_j) + \hat{Z}_{k-N_{mhe}}(z) \right\},
\]

where \( \hat{Z}_{k-N_{mhe}}(\cdot) \) is the approximate arrival cost. In principle, any function may be used as an approximation of the arrival cost, but divergence may occur due to a poor choice [31]. Therefore, estimation performance of MHE depends on the quality of \( \hat{Z}_k(\cdot) \) used, and restrictive criterion on the choice of \( \hat{Z}_k(\cdot) \) is necessary for accurate estimation.

The assumptions needed for the convergence of MHE are provided in Table 2.1 [31], where \( \gamma(\cdot) \) is continuous, strictly monotone increasing and follows the conditions:

\[
\gamma(x) \geq 0 \text{ for } x \neq 0 ,
\]

\[
\gamma(0) = 0 ,
\]

\[
\lim_{x \to \infty} \gamma(x) = \infty .
\]

Assumption 2 ensures the optimization problem has a quadratic formulation, consistent with the earlier formulation of MHE [34]. Assumptions 3 and 4 are the requirements for the approximate arrival cost. Assumption 3 roughly means that the approximate arrival cost, \( \hat{Z}_k(\cdot) \), has no better estimate than the one made by MHE.
However, the difference is bounded. Finally, Assumption 4, heuristically speaking, requires that the series of the values of $\tilde{Z}_j$ to be monotone non-increasing, and is the most restrictive assumption of MHE.

Finding an arrival cost that satisfies all these assumptions is not easy, even with the trivial solution of $\tilde{Z}_k(\cdot) = \hat{\Phi}_k$. A general framework of finding the approximate arrival cost that satisfies both Assumptions 3 and 4 has been suggested, but has also been cautioned for possibly being impractical [31]. In practice, EKF is a convenient candidate for the approximate arrival cost, but the estimate of EKF is unreliable and may have adverse effect on the estimation accuracy of MHE. A smooth estimation based on linearization of process model has also been suggested [41], and is used for part of the case studies in this dissertation.
CHAPTER 3

BAYESIAN ESTIMATION BY SEQUENTIAL MONTE CARLO SAMPLING

3.1 Background

The proposed approach relies on sequential Monte Carlo sampling (SMC) to obtain the Bayesian solution in a computationally efficient manner without relying on simplifying assumptions. Given information about the state and measurement equations and their parameters, the SMC approach only needs to select the number of samples it simulates at each time point. This approach allows the distributions to adopt any shape at each time point, making the estimates quite accurate. In addition, the sampling-based approach is recursive and does not require nonlinear or quadratic programming, as is common in Moving Horizon Based Least-Squares Estimation (MHE), thus resulting in a computationally efficient approach. These features enable SMC to outperform MHE in both accuracy and computation time. SMC is more accurate than MHE for highly skewed distributions. Otherwise, both methods are of comparable accuracy. However, SMC is found to be faster than MHE for any type of distributions. Furthermore, the Bayesian formulation provides detailed uncertainty information since the whole distribution, instead of a representative point.
like mean or mode, is estimated. This dissertation only focuses on the use of SMC for estimation in unconstrained nonlinear dynamic systems. However, these benefits are expected to be readily extended to handling constraints, gross errors, bias, and missing data.

Recently, Bayesian approaches have received much attention in the area of estimation due to the advance in Bayesian statistics and affordable, fast computation power [25]. Bayesian methods based on Monte Carlo sampling are particularly promising since accurate estimation may be achieved without the burden of investing prohibitively expensive computation like direct integration. Monte Carlo sampling based Bayesian approach has been an active area of research for a few years, and has been applied in areas like signal and image processing, state estimation, and target tracking [14, 11, 1]. These sampling based methods may be categorized by the sampling technique used, including sequential Monte Carlo sampling (SMC, [14, 2, 9, 11, 10]) and Markov chain Monte Carlo sampling (MCMC, [1, 39]). SMC draws samples from an importance function and adjusts samples’ importance with weight, while MCMC usually employs iterative algorithm for generating samples. In this dissertation, only SMC approach is explored.

In the following sections, a brief introduction to Monte Carlo sampling is provided, followed by the proposed approach of Bayesian estimation based on sequential Monte Carlo sampling. Practical issues in the application of the proposed approach is discussed, including degeneracy and slow initial convergence, and a survey of convergence of the proposed approach. Variations of sequential Monte Carlo sampling based Bayesian estimation, including one empirical Bayes approach developed by this
dissertation, proposed to alleviate the practical issues raised, comprise the last section of this chapter.

3.2 Sequential Monte Carlo Sampling for Bayesian Estimation

3.2.1 Monte Carlo Sampling

Monte Carlo sampling permits convenient computation of the properties of distributions from available samples. Expectation based on Monte Carlo sampling may be expressed as,

\[ E[f(x)] = \int f(x) \ p(x) \ dx , \quad (3.1) \]

\[ \approx \frac{1}{N} \sum_{i=1}^{N} f(x(i)) . \quad (3.2) \]

By the law of large numbers, as the number of samples goes to infinity, this estimate approaches the true value. Due to the discrete nature of Monte Carlo sampling, it is difficult to obtain the probability distribution. A crude approximation for discrete distributions, useful for building intuition, may be written as,

\[ p(x) \approx \sum_{i=1}^{N} q(i) \ \delta(x - x(i)) , \quad (3.3) \]

where \( x(i) \) is the \( i \)-th sample that approximates the distribution. The coefficient, \( q(i) \), is the probability associated with each sample. For \( x(i) \) drawn randomly from \( p(x) \), \( q(i) \) equals \( \frac{1}{N} \).

Estimation of moments based on Equation (3.2) relies on samples drawn from the true distribution, \( p(x) \). In practice, many methods are available for generating samples that follow a specified probability distribution [12, 36]. Inverse cumulative density functions (CDF) method can generate samples based on the CDF of the
probability distribution. Acceptance/Rejection method is also useful for generating samples of a probability distribution from a substitute but known distribution. This method is especially useful when direct generation of samples from a distribution is not convenient, but the value of $p(x(i))$ for a given sample, $x(i)$, is available. However, both inverse CDF and Acceptance/Rejection methods are computationally expensive. In this dissertation, importance sampling is favored for its efficiency in generating samples, as well as the fact that it does not have to rely on generating samples directly from $p(x)$.

### 3.2.2 Importance Sampling

Generating samples distributed as any probability distribution, $p(x)$, may not be convenient. However, evaluating the value of $p(x(i))$ for a given sample, $x(i)$, is usually possible. Importance sampling relaxes the requirement of generating samples from the true distribution for estimating Equation (3.1). Instead, it relies on drawing samples from a substitute distribution, $\pi(x)$, called an importance function. Importance sampling is commonly used when samples may not be easily generated from the $p(x(i))$, and may also be used in situations when $p(x(i))$ may not have a closed-form (Section 3.4.2).

Equation (3.1) may be reformulated for importance resampling as,

$$ E[f(x)] = \int f(x) p(x) \, dx , $$

$$ = \int f(x) \frac{p(x)}{\pi(x)} \pi(x) \, dx , $$

$$ \approx \frac{1}{N} \sum_{i=1}^{N} f(x(i))q^*(i) , $$
where
\[ q^*(i) = \frac{p(x(i))}{\pi(x(i))} \]  \hspace{1cm} (3.4)
is the weight function and \(x(i)\) are samples drawn from \(\pi(x)\) instead of \(p(x)\).

The following example illustrates the relevant features of importance sampling. Consider \(p(x)\) represented by the following mixture of Gaussian distributions.

\[ p(x) \sim 0.5 \, N(0, \sigma^2 = 1) + 0.5 \, N(4, \sigma^2 = 1) . \]

In this illustrative example, the first moment, or mean, is estimated via importance sampling. The following Gaussian importance function is used due to its ease of sampling,

\[ \pi(x) \sim N(0, \sigma^2 = 10) . \]

This importance function is over-supportive since it covers the domain of \(p(x)\) and has a slower decaying rate at the tail. Whenever a sample is available, the weight function, based on Equation (3.4), is easily evaluated since

\[ q^*(i) = \left\{ \frac{\frac{0.5}{\sqrt{2\pi}} e^{-0.5 (x(i))^2} + \frac{0.5}{\sqrt{2\pi}} e^{-0.5 (x(i)-4)^2}}{\sqrt{20\pi} e^{-0.5 (x(i))^2/10}} \right\} . \]

Estimation results based on 100 realizations of simulation are displayed in Table 3.1. The true mean of the non-Gaussian distribution is 2. As shown in Case I of Table 3.1, this mean is easily estimated if samples from the true distribution are available. The over-supportive importance function in Case II also does reasonably well at estimating the mean, but converges more slowly than the estimate based on the true distribution. Case III of Table 3.1 shows the estimation result based on the following under-supportive importance function,

\[ \pi(x) \sim N(0, \sigma^2 = 1) . \]
Few samples generated from this importance function fall within the high probability region of the true distribution. Even with $10^6$ samples, as shown in the last row of the last column in Table 3.1, the estimate has a terrible standard deviation despite the average estimate approaching the true value. This indicates that convergence has not been achieved. A poorer choice of importance function could result in an even worse estimate.

Proper choice of the importance function is crucial since it determines the accuracy and efficiency of estimation results. A general rule for selecting importance functions is that it should at least support the true distribution [13]. In general, importance functions whose support and shape are most similar to the true distribution tend to provide faster convergence. The relevance of good importance functions becomes evident again in Section 4.2.3. A tutorial on importance sampling provides further details and more examples [5].

In principle, Monte Carlo sampling based methods can accurately and efficiently represent and propagate distributions over time. Samples preserve all important features of distributions and hence virtually no information is lost over time. Computational load depends on both the number of samples used and the complexity of
underlying process and usually, sampling techniques require much less computation
time than nonlinear programming techniques.

3.2.3 Sequential Monte Carlo Sampling

The fundamental concept of sequential Monte Carlo sampling (SMC) follows the
same framework illustrated in Figure 2.1. This recursive Bayesian estimation is im-
plemented via Monte Carlo sampling, rather than solving the integral of Equations
(2.6) to (2.10) directly. An efficient algorithm for implementing Figure 2.1 is as fol-
 lows. At each time step, the two pieces of information required for estimating the
posterior are the samples and their corresponding weights, as in Equation (3.3). Once
samples are generated, finding the weight is as straightforward as finding the value
$q^*(i) = p(x(i))/\pi(x(i))$, which is illustrated in Section 3.2.2. SMC can also be used
when there is no closed-form of $p(x(i))$, but the weight value may be indirectly found
with proper choice of importance functions (Section 3.4.2).

In this dissertation, samples are assumed to be generated from an importance
function, $\pi(x_k|y_{1:k})$. For convenience, importance functions are selected to have a
recursive form,

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

$$
\pi(x_0) \sim p(x_0) .
$$

Therefore, the weight function may be derived as,

$$
q_k^*(i) = \frac{p(x_k(i)|y_{1:k})}{\pi(x_k(i)|y_{1:k})} ,
$$

$$
\propto \frac{p(y_k|x_k(i)) p(x_k(i)|x_{k-1}(i)) p(x_{k-1}(i)|y_{1:k-1})}{\pi(x_k(i)|x_{k-1}(i), y_k) \pi(x_{k-1}(i)|y_{1:k-1})} ,
$$

$$
\propto q_{k-1}^*(i) \frac{p(y_k|x_k(i)) p(x_k(i)|x_{k-1}(i))}{\pi(x_k(i)|x_{k-1}(i), y_k)} . \tag{3.5}
$$
• FOR times $k = 1, 2, 3, \ldots$
  — FOR samples $i = 1, 2, 3, \ldots, N$
     - Draw sample, $x_k(i)$ from an importance function, $\pi(x_k|x_{k-1}(i), y_{1:k})$
     - Assign a weight to $x_k(i)$, $q_k^*(i)$
  — END FOR
  — Normalize $q_k^*(i)$ to find $q_k(i)$
• END FOR

Table 3.2: General algorithm of SMC.

The normalized weight, $q_k^*(i)$, may then be computed as,

$$q_k(i) = \frac{q_k^*(i)}{\sum_{i=1}^{N} q_k^*(i)} .$$

Thus moments or other evaluation of the posterior can be readily computed as shown in Section 3.2.2. The SMC algorithm is represented in pseudo-code in Table 3.2 [2].

This algorithm may also be expressed in a recursive form which has prediction and update stages at each time step. In the prediction stage, samples representing the current state are generated from an importance function, $\pi(x_k|y_{1:k})$. Finding this importance function involves the multiplication of $\pi(x_k|x_{k-1}, y_k)$ and $\pi(x_{k-1}|y_{1:k-1})$ which may be interpreted as the prediction of current state based on the prior distribution, $\pi(x_{k-1}|y_{1:k-1})$. In the update stage, the prediction is updated by the information contained in the current measurement, i.e. the likelihood. Likelihood values provide a probability closeness of the prediction to the current estimate. Higher likelihood value implies that the prediction has higher probability of being the current state based on current measurement only. In SMC, information in likelihood is incorporated in Equation (3.5). Combining the prediction and the likelihood completes one time step of Bayesian estimation.
This recursive formulation also reduces the complexity of the problem compared to a moving horizon formulation. At each time step, only the prior distribution and the most current measurement is used for estimation. Since all information in previous measurements is captured in the prior distribution, the proposed SMC formulation is analogous to a moving horizon formulation with an infinite horizon. The practical challenges in using this algorithm, some solutions to these challenges and illustrative examples are discussed in the rest of this dissertation.

3.3 Practical issues

3.3.1 Degeneracy

Degeneracy is a phenomenon where after a few iterations, the weights of most samples become insignificant, while those of a few samples start to dominate. Consequently, most samples have no influence on the posterior and distributions are then determined only by a few samples. This phenomenon may weaken the successful application of Monte Carlo sampling which relies on diversity of the sample pool. Furthermore, computational resources may be wasted on samples with little or no relevance to the approximation. It may also cause spurious spikes and poor estimation.

Degeneracy may be observed by monitoring the variance of the samples’ weights, $q_k$, at every time step. In the presence of degeneracy, the variance tends to become fairly large, as depicted in Figure 3.1. This increasing variance may also imply increasing discrepancy between the true distribution and the importance function [9]. The estimation tends to be better when the importance function is closer to the true distribution.
Figure 3.1: Evolution of the variance of the weights of a linear Gaussian system. Increment of the variance may be an indication of degeneracy.
Degeneracy is more likely to occur when system or measurement noise is extremely small. In extreme case when there is no system noise, the estimated posterior may reduce to a peak and no future measurements would update this belief of the posterior. This is the stability issue of estimators and is commonly shared in many estimation techniques, including EKF [20]. Intentionally adding disturbance to the prior, or jittering [14], may reduce degeneracy created by small noise.

Degeneracy is inevitable in SMC [21] unless importance functions are selected such that the variance of samples’ weights is minimized. For example, \( p(x_k(i)|x_{k-1}(i), y_k) \), has been suggested as one such “optimal” importance function [11]. Since finding such importance functions may be difficult in general, other approaches have also been developed for reducing degeneracy. Cheng and Druzdzel [6] have suggested an adaptive algorithm for finding importance functions, which results in a more robust algorithm when unlikely measurements occur. This adaptive way of updating importance functions may be helpful since degeneracy tends to become more severe when the measurement and prediction do not match each other. Another convenient alternative is to implement resampling whenever degeneracy develops. This approach involves drawing samples from the weighted sample pool. Resampling is used in this dissertation, and is discussed in more detail in Section 3.4.1.

### 3.3.2 Slow Initial Convergence

Although the SMC approach is asymptotically convergent, in practice, a poor initial guess can cause the convergence to be very slow. Figure 3.2 illustrates a typical situation when initial convergence is likely to be slow. In this illustration, the prior and likelihood predict very different estimates of current state. Figure
3.2(a) indicates that the prior predicts the current state to be on the far left. The true likelihood for the available measurement is shown in Figure 3.2(b), and is on the far right. The weights corresponding to this likelihood and the sampling range are shown by the dashed line in Figure 3.2(a). Ideally, these weights should match the shape in Figure 3.2(b). Instead, they only capture a small and less probable portion of the true likelihood. The posterior from SMC is obtained by combining the distributions in Figure 3.2(a) and may result in a degenerate distribution, as shown by the single bar in Figure 3.2(c). The true posterior, also shown in Figure 3.2(c), has very little overlap with the posterior obtained from SMC. This indicates that the SMC estimate is not only inaccurate, but due to the the lack of overlap, it needs many more samples or time points to come closer to the true posterior. In principle, this situation is similar to Case III in Table 3.1. Many techniques may be devised for improving initial convergence such as, increasing the number of samples, performing empirical estimation of initial guess, or using an importance function which is robust to unlikely measurements. An approach based on empirical estimation is used in this dissertation, and discussed in more detail in Section 3.4.4.

### 3.3.3 Survey of Convergence Properties of Practical Methods

So far, the discussion has mainly focused on the estimation accuracy and computation efficiency. Little is done on answering other key issues of an estimator, like whether convergence is guaranteed, what is the convergence rate, or what assumptions are required for convergence. In the following, a brief survey on the convergence properties of SMC based on limiting literatures available is provided. In short, SMC
Figure 3.2: Illustration of the local support feature of sampling based techniques. (a) Prior (dark histogram) and weight (dashed line), (b) True likelihood, (c) Continuous line is the true posterior, and the histogram bar is the posterior found by SMC.
may have uniform convergence under restrictive assumptions. More theoretical work is needed to pave the foundation for any improvement of SMC.

Theoretical studies indicate that the convergence of SMC is almost sure when the importance weights, \( \{q(i)\} \), are upper bounded, a standard resampling is implemented, and the transition kernel is Feller [7]. Roughly speaking, the transition kernel is Feller implies that if there exists two samples close to each other initially, the model prediction results of these two samples will stay close. Under these weak assumptions, and when convergence is considered in terms of mean-squares error, the convergence rate is inversely proportional to the number of samples and independent of the dimensionality of the underlying problem. This implies that the number of samples may not have to be significantly increased even for high-dimensional problems. Another promising property of SMC is that when samples are drawn from the true distribution, the covariance of the estimate error may be expressed as follows [4].

\[
Q^N = (1 + N)^{-1}Q
\]

where \( Q \) is the error covariance of a true Bayesian estimate, and \( Q^N \) is the error covariance of Monte Carlo sampling based Bayesian estimate. The number of samples, \( N \), does not have to be large since \( Q^N \) quickly has only 1% deviation from the true value \( Q \) with just 100 samples.

However, SMC may require a strong assumption of being able to quickly ”forget” about its previous estimation error to have uniform convergence [7]. This assumption ensures that no error accumulation will occur over time, but does not seem to have a general and practical approach of finding such importance functions. Various practical methods exist to enhance the robustness of SMC, and may have already satisfied this condition in many problems. These variations of SMC usually improve
the quality of importance functions by incorporating the information in the current measurement. For example, an optimal importance function may be available when the noise are additive Gaussian and the measurement equation is linear with additive Gaussian noise [11]. This optimal importance function may be extended to nonlinear measurement equations by linearization. The algorithm in MCMC has also been suggested to be used in improving the quality of the importance function [3]. A hybrid approach that uses gradient decent information with SMC is also a possible venue [9]. Finally, EBSIR introduced in Section 3.4.4 is also expected to improve the quality of the importance function.

3.4 Practical Approaches

This section presents several practical methods for addressing the challenges discussed in Section 3.3. These approaches are illustrated via case studies in Section 4.1.

3.4.1 Resampling

Resampling redraws samples from a weighted sample pool based on the samples’ weights. It is a popular way of reducing the effects of degeneracy discussed in Section 3.3.1. During the resampling process, more samples are drawn in regions with higher weights and samples with insignificant weight are less likely to be drawn. Consequently, resampled samples tend to be concentrated in areas where important features exist and degeneration of samples by having few samples with high weights and many samples with low weights can be reduced or avoided. Resampling is usually implemented at the end of each time step after samples’ weights have been found.
Figure 3.3 provides a graphical illustration of resampling. In principle, samples may be drawn based both on the frequency with which they occur in the prior and the corresponding weights. However, in practice, it may be easier to expand the weighted sample pool based on weights so that during resampling, samples may be found just by one index, the adjusted frequency of occurrence in the prior. Consider \{2 4 1 3 2\} to be five samples of \(x_k(i)\) with corresponding weights, \(q_k^i = [\frac{1}{4} \frac{3}{4} 0 \frac{1}{4} \frac{1}{4}]\), respectively. The sum of the weights does not have to add to unity. In this case, the sample with value 4 has the highest weight, while the one with value 1 has a weight of zero. This difference in the relative importance of the samples is reflected via resampling by removing the sample with value 1, and tripling the number of occurrences of 4. Following this step, the expanded sample pool, is an equally weighted pool of the posterior, and is \{2 4 4 4 3 2\}.

Once the expanded sample pool is found, random drawing of samples from this pool is equivalent to random drawing of samples from a sample pool that is weighted according to the samples’ weights. For this illustration, the randomly drawn integers need to be between 1 and 6, corresponding to the sample index in the expanded sample pool. For one possible set of random integers, say \{3 6 6 3 6\}, the first resampled sample is then the third sample in the expanded sample pool, that is, a sample with value 4. Likewise, all five samples can be found and the corresponding outcome of resampling is \{4 2 2 4 2\}. Since resampling is a stochastic process, another random drawing shall result in a different outcome, but with similar statistical properties. For example, another random drawing may have random numbers of \{2 5 4 2 3\} and the corresponding resampled outcome is \{4 3 4 4 4\}. In this example, five samples are
Figure 3.3: Flow chart depicting resampling for the illustrative example.

used only for illustration purposes. In practice it is common to have a few hundred samples at least.

An alternate view of resampling is shown in Figure 3.4. In this figure, the region between zero and one is divided based on the normalized weights. As shown in the top half of Figure 3.4, samples with higher weight shall have wider regions. Similarity between the equally weighted sample pool in Figure 3.3 and the regions in Figure 3.4 is noted here. The difference is that in Figure 3.3, a random integer corresponding to the index of the expanded sample pool is required, while in Figure 3.4, a more convenient random number between zero and one is generated. That is, this algorithm in Figure 3.4 does not need to find the size of the expanded sample pool and is more convenient in practice. For example, if a random number of 0.5 is found, then the second sample is resampled since 0.5 falls into region between $\frac{1}{6}$ and $\frac{2}{3}$. This leads to the sample value of $x^*(2) = 4$. 

37
Figure 3.4: Alternate view of resampling for the illustrative example.
Resampling is not a panacea since unnecessary resampling may introduce its own challenges as samples with higher probability may be oversampled. This phenomenon is called impoverishment [2], and can be reduced by using large number of samples or implementing resampling only when necessary [21]. One way to assess the necessity of implementing resampling is by finding the effective sample size from the available samples. If the effective sample size is smaller than a chosen threshold, resampling is recommended. A practical way of evaluating the effective sample size is suggested as follows [21].

\[ \text{NESS} = \frac{1}{\sum_{i=1}^{N} q(i)^2}. \]

The value of \( \text{NESS} \) equals \( N \) when all samples have the same weight. In extreme case when only one sample has a non-zero weight, \( \text{NESS} \) equals 1. In this dissertation, a heuristic threshold value of \( \frac{1}{3N} \) is used.

3.4.2 Sampling Importance Resampling

Sampling importance resampling implements SMC with \( p(x_k|x_{k-1}) \) as the importance function, and with resampling implemented at every time step.

\[ \pi(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1}). \]  

(3.6)

This choice of importance function permits the implementation of importance sampling without the need of knowing the exact form of \( p(x_k|y_{1:k}) \). Instead, the crucial information needed, Equation (3.5), is simplified as,

\[ q_k^*(i) = q_{k-1}^*(i) \frac{p(y_k|x_k(i))}{p(y_k|x_k(i))}, \]

\[ \propto p(y_k|x_k(i)). \]  

(3.7)
FOR times $k = 1, 2, 3, \ldots$

— FOR samples $i = 1, 2, 3, \ldots, N$

  — Draw sample, $x_k(i)$ from an importance function, $p(x_k|x_{k-1}(i))$
  — Assign a weight to $x_k(i)$, $q_k(i)$ based on Equation (3.7)

— END FOR

— Implement resampling when $N_{\text{ess}} < \frac{N}{3}$

END FOR

Table 3.3: Weighted bootstrap filtering algorithm for SMC.

which depends only on the weight value at the previous time step, and the likelihood value at current time. The proportionality is because $q_k^*(i) = \frac{1}{N}$, $\forall i$, since resampling is done at every time step. This implementation is very close to the weighted bootstrap filter (WBF, [14, 2]) and is also referred to as sampling importance resampling (SIR) filter [9, 11]. In this dissertation, this importance function of $\pi(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1})$ is used since the closed-form of $p(x_k|y_{1:k})$ is usually not available, but the value of $p(y_k|x_k(i))$ can be found.

The pseudo-code for this importance function is shown in Table 3.3, and is a slight modification of the code in Table 3.2. This SMC algorithm suffers from degeneracy and slow initial convergence. The remainder of this section describes two modifications of SMC for improving its performance.

### 3.4.3 Hybrid Gradient Descent/Sampling Importance Resampling

Slow convergence of SMC may be improved by incorporating gradient descent information to develop a hybrid gradient descent SIR (HySIR) algorithm. The approach described in this section is a modification of the approach suggested by de Freitas et al. [9]. As discussed earlier in Section 3.3.2, estimation by SMC may be
bounded by available samples. However, by using the gradient descent information, samples may be able to converge sooner by moving toward the minimum of the error function. The following equation implements the hybrid approach,

\[ x_k = f_{k-1}(x_{k-1}, \omega_{k-1}) + \alpha \left( y_k - h_k(x_k, \nu_k) \right) \frac{\partial h_k(x_k, \nu_k)}{\partial x_k}, \]

(3.8)

where \( \alpha \) is a tuning parameter, and \( \frac{\partial h_k(x_k, \nu_k)}{\partial x_k} \) is the Jacobian of the measurement equation. In this dissertation, gradient descent is implemented with EKF.

The tuning parameter determines the contribution of the gradient descent. When \( \alpha = 0 \), HySIR reduces to SMC, while a large value of \( \alpha \) implies more contribution from gradient descent. However, incorporating the gradient descent, essentially the first derivative of the error function, may cause similar divergence issue as in EKF. In this dissertation, HySIR is implemented with \( \alpha = 0.7^{k-1} \) where \( k \) is the time step. The influence of gradient descent is then limited to early stages of estimation. This formulation is different from that of de Freitas et al. [9], who use a fixed value of \( \alpha \) for all \( k \). Compared to the SMC approach described in Section 3.2.3, HySIR has an extra step that shifts samples to a new location based on gradient descent. The corresponding pseudo-code is provided in Table 3.4.

De Freitas et al. [9] illustrates the superior performance of this algorithm for Bayesian neural networks. However, as illustrated in Section 4.1, the results of HySIR are mixed when used for estimation. It does improve convergence by using the gradient information, but may encounter divergence like EKF if the tuning parameter, \( \alpha \), is too large. However, if a rather small tuning parameter is used, the benefit of using gradient descent in improving slow initial convergence may not be significant as shown in CSTR study with poor initial guess. The next subsection presents a novel and more convenient approach.
Table 3.4: Hybrid gradient descent/sampling importance resampling algorithm for SMC.

3.4.4 Empirical Bayes Sampling Importance Resampling

Another way of improving initial convergence is by combining an empirical Bayes approach with SIR. The resulting Empirical Bayes SIR (EBSIR) algorithm is able to accelerate initial convergence even with a poor initial guess. As discussed in Section 3.3.2, slow initial convergence may be due to little overlap between the prediction and the likelihood. Although abundant number of samples can accelerate initial convergence, it may be impractical for online application. Finding an importance function that actually supports the posterior may not be easy in most cases, either. Instead, the EBSIR approach uses the available measurement and models to estimate the initial state at the first time step. Once a good initial distribution is obtained, the rest of the time steps rely on the SMC algorithm.

The resulting algorithm is as follows. At the first time step, the estimate is obtained with a non-informative or uniformly distributed prior. This reflects the lack of prior knowledge. In practice, this is implemented via generating many more samples at the first time step from an expanded importance function, which covers both the available initial guess and the estimate from the first measurement. The
number of samples used is much greater than those used at subsequent time steps. The estimate, $x_1$, is determined again, now with the posterior as the new prior. The resulting estimate can be a significant improvement over that obtained with a poor initial prior. This is an empirical Bayes approach since the available measurement is used to obtain the prior at $k = 1$. Such methods are widely used for practical Bayesian methods [28]. This approach requires extra computation only for the first time step, since subsequent time steps are identical to SMC. Using this approach when an accurate prior is available may cause the accuracy to deteriorate slightly. Consequently, this approach is recommended only when prior knowledge is not very good.

The only extra tuning parameter in EBSIR is the number of samples used at $k = 1$, $N_{EBSIR}$. However, this parameter is much easier to determine than the tuning parameter in HySIR. The pseudo-code for EBSIR is shown in Table 3.5.

Table 3.5: Empirical Bayes sampling importance resampling algorithm for SMC.
CHAPTER 4

BAYESIAN ESTIMATION BY SEQUENTIAL MONTE CARLO SAMPLING: APPLICATION TO UNCONSTRAINED DYNAMIC SYSTEMS

4.1 Introduction

This section presents three examples to illustrate the features of the proposed methods as compared to existing methods. The first example, a linear Gaussian system, is used to compare SMC with the optimal estimator, KF. Next is a popular nonlinear dynamic system where all distributions are highly non-Gaussian [14, 11]. This benchmark example demonstrates the benefits of SMC over MHE. Finally, an adiabatic CSTR system is studied under two different operating conditions [16, 32].

Simulation is run on a 2.0 GHz CPU with 512MB RAM personal computer using GNU/Octave, an open source implementation of Matlab. MHE is implemented by a publicly available code [15], which is considered to be robust and computationally efficient. Further improvement on the computation efficiency of MHE may not be significant since a customized optimization algorithm, SQP, is used and most parts of the code have already been compiled into fast binary format. In contrast, the SMC implementation is without compilation. It is expected that a more efficient and
compiled implementation of SMC will result in even smaller CPU times than those presented in this section.

The mean and standard deviation reported here are based on 100 realizations, with each realization consisting of $N_m = 1600$ measurements. Mean-squares error (MSE) is calculated based on the following equation.

\[
MSE = \frac{1}{N_m} \sum_{k=1}^{N_m} (\hat{x}_{k,r} - x_{k,r})^T (\hat{x}_{k,r} - x_{k,r}) ,
\]

(4.1)

where $\hat{x}_{k,r}$ is the estimate at time step $k$ of the $r$-th realization, and $x_{k,r}$ is the true state. The unit for CPU time is sec/time step. Reported results of MSE and CPU are expressed as $\mu \pm \sigma$, where $\mu$ and $\sigma$ are the mean and standard deviation of the 100 realizations.

In this dissertation, the posterior mean is chosen as the point estimate for SMC. Unfortunately, MSE often does not bring out the fact that the posterior from SMC contains much more information about the estimate than that provided by MHE or EKF. Consequently, the comparisons based on MSE in this section may be biased against SMC, and the estimation performance of MHE may be comparable to that of SMC even for non-Gaussian posteriors. Research on using a more appropriate loss function for non-Gaussian and particularly, multi-modal distributions is in progress.

4.2 Case Studies

4.2.1 A Linear Dynamic System

The process model of this linear Gaussian system [11] is,

\[
x_k = x_{k-1} + \omega_{k-1} ,
\]

(4.2)

\[
y_k = x_k + \nu_k ,
\]

(4.3)
Table 4.1: Mean-squares error and CPU load of linear Gaussian system.

<table>
<thead>
<tr>
<th></th>
<th>Y</th>
<th>KF</th>
<th>MHE</th>
<th>SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>1.0</td>
<td>0.10 ± 0.01</td>
<td>0.10 ± 0.01</td>
<td>0.11 ± 0.01</td>
</tr>
<tr>
<td>CPU Time</td>
<td>0.0007 ± 0.0</td>
<td>0.57 ± 0.01</td>
<td>0.09 ± 0.0001</td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>$m = 2$</td>
<td>$N = 500$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where $\omega_k$ and $\nu_k$ are iid Gaussian noise with $p(\omega_k) \sim N(0, \sigma^2 = 1)$ and $p(\nu_k) \sim N(0, \sigma^2 = 1)$. The initial condition for generating data is $x_0 = 0$, and the initial guess for estimation is $p(x_0) \sim N(0, \sigma^2 = 1)$.

The average MSE and CPU time are shown in Table 4.1 and Figure 4.1. All methods provide similar accuracy. As expected, KF is the fastest due to its use of closed-form solutions that are tailored for this linear Gaussian case. MHE takes the most computation despite the validity of its assumption of Gaussian distributions. This is due to the need to solve nonlinear or quadratic programming problems in each moving window, and confirms that most of the computational benefits of the Gaussian assumption vanish in MHE. Larger moving horizon for MHE is feasible in this simple problem, but the results are omitted since convergence has been achieved with horizon width 2. The proposed SMC approach is much faster than MHE and much more general than MHE and KF. This illustrates that an approach like SMC that does not rely on Gaussian approximation can be faster than approaches like MHE that do. This goes against the common notion that methods that rely on Gaussian approximation are more efficient.
4.2.2 A Nonlinear Dynamic System.

This highly nonlinear dynamic system is modeled as follows [14, 11],

\[
x_k = 0.5 x_{k-1} + \frac{25}{1 + x_{k-1}^2} x_{k-1} + 8 \cos(1.2 [k - 1]) + \omega_{k-1},
\]

\[
y_k = \frac{x_k^2}{20} + \nu_k,
\]

where \( \omega_k \) and \( \nu_k \) are iid Gaussian noises with \( p(\omega_k) \sim N(0, \sigma^2 = 10) \) and \( p(\nu_k) \sim N(0, \sigma^2 = 1) \). Initial condition for system simulation is \( x_0 = 0.1 \), while the initial guess is \( p(x_0) \sim N(0.1, \sigma^2 = 1) \). This system has been widely studied to compare various estimation methods, but has not yet been solved by MHE.

The posterior distributions shown in Figure 4.2 indicate the bimodal and skewed nature of the distributions. The average MSE and CPU time shown in Table 4.2
<table>
<thead>
<tr>
<th></th>
<th>EKF</th>
<th>MHE</th>
<th>SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>552 ± 191</td>
<td>219 ± 15</td>
<td>22 ± 2</td>
</tr>
<tr>
<td>CPU Time</td>
<td>0.0008 ± 0.0</td>
<td>0.84 ± 0.006</td>
<td>0.20 ± 0.001</td>
</tr>
<tr>
<td>Parameters</td>
<td>m = 2</td>
<td>N = 500</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Mean-squares error and CPU load of nonlinear dynamic system.

confirm the inability of EKF and MHE to handle such distributions. MHE performs better than EKF, but requires a great deal of computation. The influence of a poor prior is expected to be eased with longer moving horizons, but the computational load of the available implementation of MHE [15] becomes infeasible. In contrast, SMC outperforms both EKF and MHE with much less computation than MHE, as shown in Table 4.2 and Figure 4.3.

4.2.3 Adiabatic CSTR

A popular adiabatic CSTR is studied at two different operating conditions discussed later on. Governing equations for this system are as follows.

\[
\frac{dC}{dt} = \frac{q}{V} (C_0 - C) - k \, C \, e^{-\frac{E_A}{T}},
\]

\[
\frac{dT}{dt} = \frac{q}{V} (T_0 - T) - \frac{\Delta H}{\rho \, C_p} \, k \, C \, e^{-\frac{E_A}{T}} - \frac{U \, A}{\rho \, C_p \, V} (T - T_c),
\]

where \( C \) is the concentration, \( T \) is the temperature, \( q \) is the flow rate, \( V \) is the volume of the reactor, \( C_0 \) and \( T_0 \) are inflow concentration and temperature, \( k \, C \, e^{-\frac{E_A}{T}} \) is the reaction rate, \( \Delta H \) is the reaction heat, \( \rho \) is the density, \( C_p \) is the specific heat, \( U \) and \( A \) are the effective heat transfer coefficient and area respectively, and \( T_c \) is the temperature of the cooling fluid. The continuous differential equations are discretized via finite difference, resulting in the following model.
Figure 4.2: Evolution of the posterior of $x_1$ of nonlinear dynamic system.

Figure 4.3: Estimation performance of nonlinear dynamic system.
Table 4.3: First operating condition of unconstrained CSTR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>100</td>
<td>L/min</td>
<td>$\Delta H$</td>
<td>$-5.0 \times 10^4$</td>
<td>J/mol</td>
</tr>
<tr>
<td>$V$</td>
<td>100</td>
<td>L</td>
<td>$\rho$</td>
<td>1000</td>
<td>g/L</td>
</tr>
<tr>
<td>$C_0$</td>
<td>1.0</td>
<td>mol/L</td>
<td>$C_p$</td>
<td>0.239</td>
<td>J/g/K</td>
</tr>
<tr>
<td>$k$</td>
<td>$7.2 \times 10^{10}$</td>
<td>1/min</td>
<td>$U$</td>
<td>$5.0 \times 10^4$</td>
<td>J/cm$^2$/min/K</td>
</tr>
<tr>
<td>$E_a$</td>
<td>8750</td>
<td>K</td>
<td>$A$</td>
<td>10</td>
<td>cm$^2$</td>
</tr>
<tr>
<td>$T_0$</td>
<td>350</td>
<td>K</td>
<td>$T_c$</td>
<td>305</td>
<td>K</td>
</tr>
</tbody>
</table>

$$
x_k = \begin{bmatrix} C_k \\ T_k \end{bmatrix},
= \begin{bmatrix}
(1 - \frac{\Delta t}{V} q - \frac{\Delta t}{\rho} C_p k e^{-E_A/T}) \\
-\frac{\Delta t}{\rho} C_p k e^{-E_A/T}
\end{bmatrix} x_{k-1}
+ \begin{bmatrix}
\frac{\Delta t}{V} q C_0 \\
\frac{\Delta t}{V} U A T_0
\end{bmatrix} + \omega_{k-1},
$$

$$y_k = x_k + \nu_k,$$

where $\Delta t$ is the time span between simulation time steps.

First Operating Condition - Non-Gaussian Posteriors: This operating condition, described in Table 4.3, is studied by Henson and Seborg [16]. The normalizing factors are $C_r = 1$ mol/L and $T_r = 100$ K. Normalized initial condition is $x_0 = \begin{bmatrix} 0.5 \\ 3.5 \end{bmatrix}$, and noises in the scale of normalized variables are $p(\omega_k) \sim N (0, \sigma_\omega^2 \cdot I)$ with $\sigma_\omega^2 = 2.5 \times 10^{-7}$, and $p(\nu_k) \sim N (0, \sigma_\nu^2 \cdot I)$ with $\sigma_\nu^2 = 0.0025$.

Figure 1.1 shows the evolution of the posterior, indicating skewed and multimodal distributions. Unfortunately, the ability of SMC to capture non-Gaussian distributions is not illustrated very well by the point estimates used in this example due to the reasons discussed at the beginning of this section. Two initial guesses are
studied in this dissertation, a perfect initial guess and an extremely poor one. The reported MSE is averaged over both variables.

For the *perfect initial guess*, increasing the number of samples improves the accuracy of SMC while increasing the computational cost (Table 4.4 and Figure 4.5). HySIR and EBSIR both show performance comparable to SMC for both, estimation error and computation load.

A *poor initial guess* is common in many practical situations, and may cause SMC and other methods to exhibit slow initial convergence and poor accuracy, as shown in the second column of Table 4.5. However, once SMC converges, its MSE performance is similar to that of EKF and MHE, as shown in the third column of Table 4.5. This result is also illustrated in Figure 4.6, which shows that, in the beginning, SMC does not converge as fast as other approaches. However, once convergence is achieved around $k = 200$, SMC follows the true state much more closely than EKF or MHE. As shown in Table 4.5, HySIR improves the slow initial convergence that SMC faces

<table>
<thead>
<tr>
<th></th>
<th>MSE/\sigma_\nu^2</th>
<th>CPU time</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Y</strong></td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>0.14 ± 0.02</td>
<td>0.0005 ± 0.0</td>
<td></td>
</tr>
<tr>
<td>MHE</td>
<td>0.02 ± 0.01</td>
<td>0.17 ± 0.005</td>
<td>(m = 2)</td>
</tr>
<tr>
<td></td>
<td>0.02 ± 0.01</td>
<td>0.27 ± 0.004</td>
<td>(m = 5)</td>
</tr>
<tr>
<td></td>
<td>0.02 ± 0.01</td>
<td>0.46 ± 0.01</td>
<td>(m = 10)</td>
</tr>
<tr>
<td>SMC</td>
<td>0.05 ± 0.04</td>
<td>0.05 ± 0.0004</td>
<td>(N = 500)</td>
</tr>
<tr>
<td></td>
<td>0.03 ± 0.04</td>
<td>0.13 ± 0.0006</td>
<td>(N = 1000)</td>
</tr>
<tr>
<td>HySIR</td>
<td>0.03 ± 0.04</td>
<td>0.13 ± 0.0006</td>
<td>(N = 1000), (\alpha = 0.7^{k-1})</td>
</tr>
<tr>
<td>EBSIR</td>
<td>0.05 ± 0.12</td>
<td>0.13 ± 0.0005</td>
<td>(N = 1000), (N_{EBSIR} = 5000)</td>
</tr>
</tbody>
</table>

Table 4.4: Mean-squares error and CPU load of unconstrained CSTR for first operating condition with perfect initial guess.
Figure 4.4: Estimation performance of unconstrained CSTR for first operating condition with perfect initial guess: (a) Concentration (b) Temperature.

Figure 4.5: Estimation performance of unconstrained CSTR for first operating condition with perfect initial guess.
Table 4.5: Mean-squares error and CPU load of unconstrained CSTR for first operating condition with poor initial guess.

<table>
<thead>
<tr>
<th>Method</th>
<th>$MSE/\sigma^2$</th>
<th>CPU time</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>$0.48 \pm 0.04$</td>
<td>$0.14 \pm 0.03$</td>
<td>$0.0005 \pm 0.0$</td>
</tr>
<tr>
<td>MHE</td>
<td>$0.34 \pm 0.03$</td>
<td>$0.02 \pm 0.01$</td>
<td>$0.17 \pm 0.004$</td>
</tr>
<tr>
<td></td>
<td>$0.33 \pm 0.07$</td>
<td>$0.02 \pm 0.01$</td>
<td>$0.28 \pm 0.06$</td>
</tr>
<tr>
<td></td>
<td>$0.34 \pm 0.03$</td>
<td>$0.02 \pm 0.01$</td>
<td>$0.46 \pm 0.01$</td>
</tr>
<tr>
<td>SMC</td>
<td>$1.73 \pm 3.44$</td>
<td>$0.03 \pm 0.04$</td>
<td>$0.04 \pm 0.0002$</td>
</tr>
<tr>
<td></td>
<td>$1.68 \pm 4.05$</td>
<td>$0.03 \pm 0.03$</td>
<td>$0.13 \pm 0.0005$</td>
</tr>
<tr>
<td>HySIR</td>
<td>$0.20 \pm 0.81$</td>
<td>$0.03 \pm 0.02$</td>
<td>$0.05 \pm 0.0001$</td>
</tr>
<tr>
<td></td>
<td>$0.04 \pm 0.05$</td>
<td>$0.02 \pm 0.01$</td>
<td>$0.13 \pm 0.0004$</td>
</tr>
<tr>
<td>EBSIR</td>
<td>$0.09 \pm 0.23$</td>
<td>$0.03 \pm 0.03$</td>
<td>$0.05 \pm 0.0002$</td>
</tr>
<tr>
<td></td>
<td>$0.03 \pm 0.02$</td>
<td>$0.02 \pm 0.02$</td>
<td>$0.13 \pm 0.0005$</td>
</tr>
</tbody>
</table>

$\alpha = 0.7^{k-1}$ for both cases of HySIR, and $N_{EBSIR} = 5000$ for both cases of EBSIR.

while EBSIR exhibits even better performance. The performance of HySIR may be improved by modifying the tuning parameter, $\alpha$, so that the gradient descent may have more influence in the beginning. However, tuning HySIR to bring it closer to EKF may not be favorable if highly non-Gaussian distributions are present. In contrast to HySIR, EBSIR needs hardly any tuning except for deciding the number of samples for $k = 1$, and shows the best overall performance.

Second Operating Condition - Nearly Gaussian Distributions: This operating condition in Table 4.6 is studied by Jang et al. [19], Liebman et al. [24], and Robertson and Lee [32]. Figure 4.8 shows the evolution of the posterior which exhibits Gaussian-like unimodal distributions most of the time. In cases when the posterior is almost Gaussian, SMC is not expected to be much more accurate than a well-tuned MHE, but can be much faster.
Figure 4.6: Illustration of slow initial convergence of the concentration of unconstrained CSTR with poor initial guess.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>10</td>
<td>cc/s</td>
<td>$\Delta H$</td>
<td>-27000</td>
<td>cal/mol</td>
</tr>
<tr>
<td>$V$</td>
<td>1000</td>
<td>cc</td>
<td>$\rho$</td>
<td>0.001</td>
<td>g/cc</td>
</tr>
<tr>
<td>$C_0$</td>
<td>$6.5 \times 10^{-6}$</td>
<td>mol/cc</td>
<td>$C_p$</td>
<td>1.0</td>
<td>cal/g/K</td>
</tr>
<tr>
<td>$k$</td>
<td>$7.86 \times 10^{12}$</td>
<td>1/s</td>
<td>$U$</td>
<td>$5.0 \times 10^{-4}$</td>
<td>cal/cm$^2$/s/K</td>
</tr>
<tr>
<td>$E_a$</td>
<td>14090</td>
<td>K</td>
<td>$A$</td>
<td>10</td>
<td>cm$^2$</td>
</tr>
<tr>
<td>$T_0$</td>
<td>350</td>
<td>K</td>
<td>$T_c$</td>
<td>340</td>
<td>K</td>
</tr>
</tbody>
</table>

Table 4.6: Second operating condition of unconstrained CSTR.
Figure 4.7: Estimation performance of unconstrained CSTR for first operating condition with poor initial guess. (a) Overall MSE. (b) MSE after initial dynamics. (c) CPU load.
Figure 4.8: Evolution of the posterior of the concentration of unconstrained CSTR for the second operating condition.

The normalizing factors are $C_r = 10^{-7}$ mol/cc and $T_r = 100$ K. Normalized initial condition is $x_0 = \begin{bmatrix} 1.53 \\ 4.61 \end{bmatrix}$, and noises in the scale of normalized state are, $p(\omega_k) \sim N(0, \sigma_\omega^2 \cdot I)$ with $\sigma_\omega^2 = 0.0005$ and $p(\nu_k) \sim N(0, \sigma_\nu^2 \cdot I)$ with $\sigma_\nu^2 = 0.05$. As shown in Table 4.7, all NDDR methods have similar accuracy. However, SMC and its modifications are much more efficient than MHE.

4.3 Summary

This dissertation introduces a novel approach for estimation or rectification in nonlinear dynamic process systems. This approach is based on a rigorous Bayesian formulation that uses sequential Monte Carlo (SMC) sampling to propagate all information recursively, while minimizing assumptions about the system. The resulting
Table 4.7: Mean-squares error and CPU load of unconstrained CSTR for second operating condition.

<table>
<thead>
<tr>
<th></th>
<th>MSE/σ²&lt;sub&gt;y&lt;/sub&gt;</th>
<th>CPU time</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y 1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>0.034 ± 0.004</td>
<td>0.0005 ± 0.0</td>
<td></td>
</tr>
<tr>
<td>MHE</td>
<td>0.033 ± 0.004</td>
<td>0.47 ± 0.0003</td>
<td>m = 2</td>
</tr>
<tr>
<td>SMC</td>
<td>0.033 ± 0.004</td>
<td>0.04 ± 0.0002</td>
<td>N = 500</td>
</tr>
<tr>
<td>HySIR</td>
<td>0.033 ± 0.004</td>
<td>0.04 ± 0.0002</td>
<td>N = 500, α = 0.7&lt;sup&gt;k−1&lt;/sup&gt;</td>
</tr>
<tr>
<td>EBSIR</td>
<td>0.034 ± 0.004</td>
<td>0.04 ± 0.0002</td>
<td>N = 500, N&lt;sub&gt;EBSIR&lt;/sub&gt; = 5000</td>
</tr>
</tbody>
</table>

Figure 4.9: Estimation performance of unconstrained CSTR for second operating condition.
approach does not rely on common assumptions of Gaussian or fixed-shape distributions, which are readily violated in nonlinear dynamic systems. Distributions at each time step are represented by samples whose values are propagated efficiently through the system and measurement equations, while capturing prior knowledge. The recursive nature of SMC without the need for solving constrained nonlinear programming problems makes the proposed approach more computationally efficient than MHE, while providing similar or better estimation accuracy. Illustrative examples show that SMC can outperform current NDDR approaches, including EKF and MHE. SMC is always found to be more computationally efficient than MHE even when MHE’s assumption of Gaussian distributions is satisfied. More efficient implementation of SMC is possible and should further enhance its computational benefits. The accuracy of SMC is comparable to EKF and MHE for Gaussian distributions, but much better for non-Gaussian distributions. However, better error metrics need to be devised to highlight the benefits of having more information about the posterior distribution in SMC as compared to EKF or MHE.
CHAPTER 5

BAYESIAN ESTIMATION BY SEQUENTIAL MONTE CARLO SAMPLING: APPLICATION TO CONSTRAINED DYNAMIC SYSTEMS

5.1 Introduction

Process constraints commonly exist in engineering operation processes. Constraints may be roughly categorized as physical constraints, and model constraints. Physical constraints include conditions that based on physical laws, like the volume or mass of a physical material has to be non-negative. Model constraints include constraints based on model relationship. For example, the model based on mass balance can be a model constraint.

Enforcement of process constraints essentially assigns zero probability to regions where process constraints are violated. This implies that the resulting distribution is usually truncated and non-Gaussian. Existing methods are expected to have inferior performance when constraints exist because these methods usually rely on assumptions about the nature of the model or the probability distributions of the underlying variables to obtain a tractable optimization problem (Section 2.3). Distributions shown in Figure 5.1 are generated from a popular continuously stirred tank reactor (CSTR) case study when a non-negative constraint is enforced on state variables.
Figure 5.1: Evolution of the posterior of the concentration of constrained CSTR.

[19, 24, 16, 34]. This figure is generated based on the estimate of the proposed Bayesian method with 5000 samples. Bayesian estimation handles truncated distributions naturally without the necessity of approximating the underlying distributions with Gaussian or other fixed-shape distributions. The discrepancy between the predetermined shape with the true distributions are well-known and have been widely recognized [30, 33]. Nevertheless, these assumptions are popular since they may permit existing methods to solve a convenient problem instead of the actual estimation problem.

In this chapter, a general framework for handling constrained estimation problem is proposed. Bayesian estimation for unconstrained problems by SMC has been proposed in many areas such as signal and image processing, target recognition and process engineering operations [14, 9, 11]. However, this is the first time a practical
algorithm is proposed for Bayesian estimation into constrained estimation problem. An acceptance/rejection algorithm is used to enforce process constraints. Similar to the previous chapter, performance of the proposed algorithm is compared with existing methods, including EKF and MHE, on selected case studies. Results of case studies confirm the expectation that a method without relying on simplification of the real problem has better estimation accuracy.

In the following sections, discussion of the solution strategy when constraints exist is first provided. Performance of the proposed approach is compared with existing approaches in the case study section.

5.1.1 Bayesian Estimation with Constraints

This advantage of Bayesian estimation over existing methods come from the fact that Bayesian estimation considers all variables to be stochastic, and allows distributions to adopt their shapes according to system dynamics, process constraints, and measurements. Therefore, assigning zero probability on parts of the Bayesian estimate is straightforward, and may be implemented by a verification step. In fact, solution strategy for unconstrained Bayesian estimation is expected to be applicable to constrained cases with little modification. In the following, a brief introduction on how Bayesian estimation may handle constraints is provided.

Process constraints may be applied on state variables or measurements. For constraints on state variables, $x_k$, enforcement can be implemented by assigning zero probability for certain $x^*$, where process constraints are violated. For constraints on measurements, $y_k$, enforcement may be implemented by assigning zero value to the likelihood distribution. Enforcement of process constraints may raise the concern of
causal correlation between state variables and disturbances [30]. For example, enforcing constraints on state variables may build up a correlation between $x_k$ and $\omega_k$ since $f(x_k, \omega_k)$ need to be non-equal to $x_{k+1}^*$. In this dissertation, causal correlation is assumed to be insignificant and no correlation exists between state variables and disturbances.

Constraints on state variables may only need to be enforced at time step one for the following trivial system,

$$x_k = x_{k-1}.$$  

The reason is that after time step one, $p(x_k = x^*|y_{1:k})$ will automatically be zero. That is,

$$p(x_k = x^*|y_{1:k}) \propto p(y_k|x_k = x^*)p(x_k = x^*|y_{1:k-1}),$$

and

$$p(x_k = x^*|y_{1:k-1}) = \int p(x_k = x^*|x_{k-1})p(x_{k-1}|y_{1:k-1}) \, dx_{k-1}$$

$$= \int p(x_k = x^*|x_{k-1} = x^*)p(x_{k-1} = x^*|y_{1:k-1}) \, dx_{k-1}$$

$$= 0$$

**SMC and Enforcement of Constraints**

Enforcement of process constraints in SMC is also straightforward and can be achieved by keeping samples that satisfy constraints. In practice, this can be implemented by an acceptance/rejection algorithm. More computation is expected to account for this extra step for enforcing constraints, but the increment is also expected to be minor. For instance, samples will be generated from a candidate importance function and only the ones that satisfy process constraints will be kept and further
Table 5.1: Algorithm of SMC with enforcement of constraints.

processed. For practical convenience, a slightly more than required number of samples will be generated, so that a fixed number of samples may be propagated for all time steps. Usually, this number does not have to be large, since prior samples are already satisfying the constraints. For constraints on the measurements, the enforcement of constraints may be implemented by assigning zero values to the weight function, \( q(i) \).

The algorithm used in this dissertation follows the same formulation proposed in Table 3.3, and may be represented as the pseudo-code in Table 5.1. Compared to the unconstrained formulation, an extra step of enforcing constraints by an acceptance/rejection algorithm is added.

5.2 Case Studies

Two dynamic systems are studied in this dissertation, a linear dynamic system and a typical chemical engineering problem, continuously stirred tank reactor (CSTR). The linear system has a constraint on the system noise, which essentially results in a constrained estimation problem with non-Gaussian noise [33]. The CSTR case is a well-studied nonlinear dynamic problem [19, 16, 34], and two conditions are
considered, including one with Gaussian measurement noise, and the other with non-Gaussian measurement noise. SMC shows significant improvement on estimation accuracy over existing methods especially when distributions exhibit non-Gaussian features. In addition, SMC consistently requires less computation than MHE in nonlinear dynamic systems.

Simulation is run on a 2.0 GHz CPU with 512MB RAM personal computer using GNU/Octave, an open source implementation of Matlab. Same as the previous chapter, MHE is implemented by a publicly available binary code [15], while the SMC implementation is without compilation. It is expected that a more efficient and compiled implementation of SMC will result in even smaller CPU times than those presented in this section. Performance is also compared based on MSE (Equation 4.1) and CPU consumption as in the previous chapter. As mentioned earlier, MSE may not be a fair indicator for estimation accuracy, and more appropriate index metrics are under development.

Area of **high probability density (HPD)** are highlighted in each case study to illustrate the potential benefits of having the whole posterior as the estimate compared to merely point estimates as the estimation results. Contrary to most estimation methods, SMC can easily provide much more information of the posterior. When the posterior exhibits multi-modal, skewed features, SMC can be advantageous over existing methods in providing not just mean, mode or median, but also multiple modes, skewness, or other useful information.

Boundaries of the HPD regions can be found based on the following equations,

\[
\frac{100\% - \theta}{2} = \int_{-\infty}^{x^{LL}} p(x_k|y_{1:k}) \, dx_k ,
\]
\[
\frac{100\% + \theta}{2} = \int_{-\infty}^{x_{UL}} p(x_k | y_{1:k}) \, dx_k ,
\]

where \( x_{LL} \) is the lower bound of the HPD, and \( x_{UL} \) is the upper bound of HPD. The value \( \theta \) is the coverage ratio of the HPD. A practical way of estimating these limits is to sort the samples based on their state values, and then pick the samples that correspond to the chosen coverage ratio. In this dissertation, boundary points of the 95% coverage of the HPD are selected.

5.2.1 A Constrained Linear Dynamic System

A linear Gaussian system with non-Gaussian error is first studied. The process model is provided in the following equations [30],

\[
x_k = \begin{bmatrix} 0.9962 & 0.1949 \\ -0.1949 & 0.3815 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0.03393 \\ 0.1949 \end{bmatrix} \omega_{k-1} ,
\]

\[
y_k = [1 - 3] x_k + \nu_k ,
\]

where \( \{\omega_k\} \) and \( \{\nu_k\} \) are iid Gaussian noise with \( p(\omega_k) \sim N(0, \sigma^2 = 1) \) and \( p(\nu_k) \sim N(0, \sigma^2 = 0.1) \). The system noise is non-Gaussian with a non-zero mean (Figure 5.2). Simulation of the true state is based on a Gaussian initial state of \( p(x_0) \sim N(0, \sigma^2 = 1) \). Initial guess of estimation methods have the same distribution as the initial state. There are 50 measurements in each realization. A non-negative constraint is enforced on the system noise, as provided in the following equation.

\[
\omega_{k-1} \geq 0 .
\]

SMC exhibits superior estimation accuracy over MHE and EKF. EKF, essentially Kalman filtering in this problem, fails to provide accurate estimation since no process
constraint is enforced. MHE has better estimation accuracy than EKF, since MHE can handle constraint. However, MHE performs much worse than SMC mainly because MHE uses a poor choice of the approximate arrival cost, EKF. As illustrated in Figure 5.4, EKF is far away from the true state. MHE may improve its estimation accuracy by adopting a better approximate of the arrival cost, but is still expected to have inferior estimation than SMC for its invalid Gaussian assumption on the prior. Contrary to EKF and MHE, SMC benefits from not making presumption on distributions, thus can easily handles constraints and non-Gaussian distributions. Despite its inadequate estimation accuracy, EKF benefits from its closed-form solution and has the most efficient algorithm. In this linear problem, MHE shows less consumed
<table>
<thead>
<tr>
<th>Method</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>CPU time</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>0.51 ± 0.12</td>
<td>0.05 ± 0.01</td>
<td>0.0005 ± 0.0</td>
<td></td>
</tr>
<tr>
<td>MHE</td>
<td>0.21 ± 0.06</td>
<td>0.02 ± 0.006</td>
<td>0.02 ± 0.0002</td>
<td>$m = 2$</td>
</tr>
<tr>
<td></td>
<td>0.14 ± 0.05</td>
<td>0.02 ± 0.005</td>
<td>0.04 ± 0.0004</td>
<td>$m = 5$</td>
</tr>
<tr>
<td></td>
<td>0.14 ± 0.05</td>
<td>0.02 ± 0.005</td>
<td>0.07 ± 0.0006</td>
<td>$m = 10$</td>
</tr>
<tr>
<td>SMC</td>
<td>0.10 ± 0.41</td>
<td>0.05 ± 0.01</td>
<td>0.01 ± 0.0002</td>
<td>$N = 200$</td>
</tr>
<tr>
<td></td>
<td>0.04 ± 0.03</td>
<td>0.01 ± 0.002</td>
<td>0.05 ± 0.0002</td>
<td>$N = 500$</td>
</tr>
<tr>
<td></td>
<td>0.04 ± 0.02</td>
<td>0.01 ± 0.003</td>
<td>0.24 ± 0.002</td>
<td>$N = 1500$</td>
</tr>
</tbody>
</table>

Table 5.2: Mean-squares error and CPU load of constrained linear system.

CPU time than SMC, and may be explained by using a special algorithm for linear dynamic systems and uses compiled code.

5.2.2 A Constrained Adiabatic CSTR

In this section, a popular process engineering problem, an adiabatic CSTR, is studied. Governing equations are provided as follows.

Two conditions are studied for this system, including one with Gaussian measurement noise, and the other with non-Gaussian noise. Common operating conditions and parameters of these two cases can be found in Section 4.2.3. A non-negative constraint is enforced on state variables on both situations, and is provided in the following equation,

$$x_k \geq 0.$$  

Constrained CSTR Case with Gaussian Noise

In this case study, the system noise in the scale of the normalized state variables is $p(\omega) \sim N(0, \sigma_\omega^2 \cdot I)$, where $\sigma_\omega^2 = 2.5 \times 10^{-7}$, and the measurement noise is $p(\nu) \sim N(0, \sigma_\nu^2 \cdot I)$, where $\sigma_\nu^2 = 0.0025$. The initial guess has the same distribution
Figure 5.3: Estimation performance of constrained linear system.

Figure 5.4: Rectification result of $x_1$ of constrained linear system.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Concentration</th>
<th>Normalized Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>$1.7 \times 10^{-4} \pm 6.7 \times 10^{-5}$</td>
<td>$6.3 \times 10^{-4} \pm 2.8 \times 10^{-4}$</td>
</tr>
<tr>
<td>MHE</td>
<td>$4.0 \times 10^{-6} \pm 2.1 \times 10^{-6}$</td>
<td>$1.1 \times 10^{-4} \pm 4.3 \times 10^{-5}$ $m = 2$</td>
</tr>
<tr>
<td></td>
<td>$3.9 \times 10^{-5} \pm 2.2 \times 10^{-5}$</td>
<td>$1.0 \times 10^{-4} \pm 4.7 \times 10^{-5}$ $m = 5$</td>
</tr>
<tr>
<td></td>
<td>$3.7 \times 10^{-5} \pm 2.0 \times 10^{-5}$</td>
<td>$1.1 \times 10^{-4} \pm 5.0 \times 10^{-5}$ $m = 10$</td>
</tr>
<tr>
<td>SMC</td>
<td>$8.3 \times 10^{-5} \pm 4.1 \times 10^{-4}$</td>
<td>$1.4 \times 10^{-4} \pm 1.4 \times 10^{-3} N = 500$</td>
</tr>
<tr>
<td></td>
<td>$3.9 \times 10^{-5} \pm 2.3 \times 10^{-5}$</td>
<td>$1.1 \times 10^{-4} \pm 3.5 \times 10^{-4} N = 1000$</td>
</tr>
<tr>
<td></td>
<td>$3.8 \times 10^{-5} \pm 2.2 \times 10^{-5}$</td>
<td>$1.1 \times 10^{-4} \pm 5.6 \times 10^{-5} N = 2000$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU time</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>$0.001 \pm 0.0$</td>
</tr>
<tr>
<td>MHE</td>
<td>$0.2 \pm 0.01$ $m = 2$</td>
</tr>
<tr>
<td></td>
<td>$0.4 \pm 0.01$ $m = 5$</td>
</tr>
<tr>
<td></td>
<td>$0.6 \pm 0.02$ $m = 10$</td>
</tr>
<tr>
<td>SMC</td>
<td>$0.02 \pm 0.002 N = 500$</td>
</tr>
<tr>
<td></td>
<td>$0.07 \pm 0.002 N = 1000$</td>
</tr>
<tr>
<td></td>
<td>$0.17 \pm 0.01 N = 2000$</td>
</tr>
</tbody>
</table>

Table 5.3: Mean-squares error and CPU load of constrained CSTR with Gaussian noise.

as the initial state as $p(x_0) \sim \mathcal{N}(\mu_{x_0}, \sigma_{x_0}^2)$, where $\mu_{x_0} = [0.5; 350]$. There are 400 measurements in each realization.

SMC consistently requires less computation load than MHE even when constraints are enforced. However, SMC has comparable accuracy with MHE, but is much better than EKF (Table 5.3, Figure 5.5). EKF has poor estimation performance due to poor estimate by linearization and without enforcing constraints. Unlike the linear case study, MHE uses a smooth estimate for the approximate arrival cost [41], thus has competitive MSE with SMC. Unfortunately, the ability of SMC to capture non-Gaussian distributions is not illustrated by the point estimates, and MSE may not
Figure 5.5: Estimation performance of constrained CSTR with Gaussian noise.

provide the insight that SMC can provide more information since the whole distribution, not just a point estimate, is available. A proper estimation measure is under development.

Constrained CSTR Case with Non-Gaussian Noise

The same CSTR system but with skewed measurement noise is studied in this section. Skewed distribution is not uncommon in real data from industries, and in this dissertation, measurement noise is assumed to be perfectly approximated by double half normal distributions [33]. The probability function for noise is provided in the following equations,

\[
p(\nu_1) \begin{cases} 
    w_1 \exp\left(-\frac{1}{2} \frac{(\nu_1-\mu_1)^2}{\sigma_{1,1}}\right) & -\infty \leq \nu_1 \leq \mu_1 \\
    w_2 \exp\left(-\frac{1}{2} \frac{(\nu_1-\mu_1)^2}{\sigma_{2,1}}\right) & \mu_1 \leq \nu_1 \leq \infty 
\end{cases}
\]
Figure 5.6: Rectification result of the concentration of constrained CSTR with Gaussian noise.

Figure 5.7: Rectification result of the temperature of constrained CSTR with Gaussian noise.
\[
\begin{align*}
\mu_{\nu_1} &= 0 \quad \sigma_{\mu_{1,1}}^2 = 0.21 \quad \sigma_{\nu_1}^2 = 2.1 \\
\mu_{\nu_2} &= 0 \quad \sigma_{\mu_{1,2}}^2 = 0.05 \quad \sigma_{\nu_2}^2 = 2.43
\end{align*}
\]

Table 5.4: Parameters of double-half normal distributions of non-Gaussian noise.

\[p(\nu) = 0.25 \exp\left(-\frac{(\nu_2 - \mu_{\nu_2})^2}{2\sigma_{\nu_2}^2}\right) -\infty \leq \nu_2 \leq \mu_{\nu_2} ,
\]

\[p(\nu_{2}) \quad \left\{ \begin{array}{l}
w_3 \exp\left(-\frac{1}{2} \frac{(\nu_2 - \mu_{\nu_2})^2}{\sigma_{\nu_2}^2}\right) -\infty \leq \nu_2 \leq \mu_{\nu_2} , \\
w_4 \exp\left(-\frac{1}{2} \frac{(\nu_2 - \mu_{\nu_2})^2}{\sigma_{\nu_2}^2}\right) \mu_{\nu_2} \leq \nu_2 \leq \infty \end{array} \right.,
\]

where \(p(\nu_1)\) is the noise for the concentration, and \(p(\nu_2)\) is the noise of the temperature. Parameters for these two double-half normal noise are provided in Table 5.4, and \(p(\nu_1)\) is illustrated in Figure 5.8. Normalization parameters, \(\{\omega_i\}\), are chosen so that the two half normal distribution in a double-half normal distribution have the same probability value at the interface point. Similar to previous cases, the initial
guess has the same distribution as the initial state, \( p(x_0) \sim N(\mu_{x_0}, \sigma^2_{x_0}) \). There are 600 measurements in each realization.

SMC handles the non-Gaussian measurement noise without the necessity of changing the solution formulation, and updating the weight via the likelihood value by Equations \( p(\nu_1) \) and \( p(\nu_2) \). On the contrary, MHE has to modify its solution algorithm to be able to accommodate the extra least-squares terms introduced by Gaussian sum approximation. In this dissertation, MHE with Gaussian sum approximation is not implemented, and instead, Gaussian approximation of non-Gaussian noise is used. The approximate Gaussian distribution has inaccurate mean as zero, but exact variance based on the follow formula \[33\],

\[
E[\sigma^2_{\nu_i}] = \frac{\sigma_{\nu_1,i}^3 + \sigma_{\nu_2,i}^3}{\sigma_{\nu_1,i}^3 + \sigma_{\nu_2,i}^3} - \frac{2(\sigma_{\nu_2,i}^2 - \sigma_{\nu_1,i}^2)}{\pi(\sigma_{\nu_1,i}^2 + \sigma_{\nu_2,i}^2)^2}.
\]

SMC exhibits significant improvement on estimation accuracy over EKF and MHE (Table 5.5 and Figure 5.9). This is not a surprise since both EKF and MHE uses inferior information on non-Gaussian measurement disturbances.

SMC shows broaden marginal confidence interval when the system changes its trend (Figure 5.10). This phenomenon may be explained as the two state variables have high correlation, and the resulting posterior is concentrated in a tilted, corn-shaped region which produces an artifact impression of divergence (right plot in Figure 5.12).

### 5.3 Conclusion and Discussions

In this chapter, a practical approach for extending the application of sequential Monte Carlo sampling based Bayesian estimation into constrained problems is developed. This algorithm is based on the previous work for unconstrained estimation
<table>
<thead>
<tr>
<th></th>
<th>MSE</th>
<th>CPU time</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Concentration</td>
<td>Normalized Temperature</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>1.64</td>
<td>2.14</td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>0.28 ± 0.04</td>
<td>0.36 ± 0.04</td>
<td>0.001 ± 0.0</td>
</tr>
<tr>
<td>m = 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MHE</td>
<td>0.01 ± 0.01</td>
<td>0.02 ± 0.01</td>
<td>0.2 ± 0.01</td>
</tr>
<tr>
<td>m = 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.01 ± 0.01</td>
<td>0.02 ± 0.01</td>
<td>0.4 ± 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>m = 10</td>
</tr>
<tr>
<td></td>
<td>0.01 ± 0.01</td>
<td>0.02 ± 0.01</td>
<td>0.6 ± 0.02</td>
</tr>
<tr>
<td>SMC</td>
<td>0.001 ± 0.0005</td>
<td>0.002 ± 0.001</td>
<td>0.04 ± 0.0002</td>
</tr>
<tr>
<td>N = 500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.001 ± 0.0005</td>
<td>0.002 ± 0.001</td>
<td>0.2 ± 0.001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N = 1000</td>
</tr>
<tr>
<td></td>
<td>0.001 ± 0.0005</td>
<td>0.002 ± 0.001</td>
<td>0.4 ± 0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N = 2000</td>
</tr>
</tbody>
</table>

Table 5.5: Mean-squares error and CPU load of constrained CSTR with non-Gaussian noise.

Figure 5.9: Estimation performance of constrained CSTR with non-Gaussian noise.
Figure 5.10: Rectification result of the concentration of constrained CSTR with non-Gaussian noise.

Figure 5.11: Rectification result of the temperature of constrained CSTR with non-Gaussian noise.
problem for nonlinear dynamic systems (Chapter 4), and enforces constraints via an acceptance/rejection algorithm. This formulation is general and alternative ways of constructing importance functions or enforcing constraints can be easily applied. Results from case studies indicate that SMC is capable of handling constraints and provide accurate estimate with efficient computation. SMC easily outperforms EKF in estimation accuracy since EKF does not enforce constraints and relies on linearization for estimation. SMC also shows significant improvement on estimation accuracy over MHE especially when distributions exhibit non-Gaussian features. MHE may improve its estimation accuracy via Gaussian sum approximation to handle non-Gaussian distributions, but may not have a practical algorithm in general. In short, SMC handles constraints and non-Gaussian distributions naturally without the necessity of approximating the underlying distributions against convenient ones. In terms of computation efficiency, SMC consistently requires less CPU time than MHE.
especially in nonlinear dynamic systems. This advantage is expected to be further extended when binary code is used or parallel computation is implemented.

A short survey on the convergence properties of existing methods is also provided in this chapter. Both SMC and MHE may have uniform convergence under restrictive assumptions. MHE may have a general formulation with guaranteed convergence, but may not have a practical algorithm for such formulation. On the other hand, SMC may have an convergence rate inversely proportional to the number of samples, but may not have uniform convergence. These conclusions are based on limited literature available, and more investigation is required to further progress the understating of these practical methods.
CHAPTER 6

BAYESIAN ESTIMATION BY SEQUENTIAL MONTE CARLO SAMPLING: COMPUTATION LOAD VS DIMENSIONALITY

SMC has been successfully applied, and illustrated promising improvement on estimation performance in various research disciplines, including target tracking, signal processing, and state estimation [14, 2, 9, 11, 10]. In previous chapters, improvement on computation load over MHE has also been reported (Chapter 4, and 5). However, these illustrations were based on relatively low-dimensional examples, and were not enough to address skepticism about the feasibility and advantages of SMC based Bayesian estimation for higher dimensional problems. Such skepticism is well-founded for data-based methods due to the “curse of dimensionality”, and may also apply to sampling based methods. Traditionally, Bayesian methods have not been feasible for high-dimensional systems since these methods usually require integration in high-dimensional space over a relatively fine grid, which leads to formidable requirement of computation. This experience leads to the perception and expectation that the computation load of sampling based methods might also increase exponentially thus making them impractical for most large scale problems.
This chapter applies our previous work on Bayesian rectification by SMC to large scale nonlinear dynamic systems and compares the computational efficiency and accuracy with MHE and EKF. The results indicate that Bayesian estimation by SMC is feasible even for high-dimensional systems, a 8-D polymerization reactor. The selected process model is nonlinear, dynamic and involves eight state variables. Non-negativity constraints are also imposed on some variables. The estimation results show that SMC can be significantly faster than MHE for similar estimation accuracy. Since the posterior distributions for the selected operating conditions may be approximated quite well as Gaussian distributions, the accuracy of SMC and MHE can be comparable.

The rest of this chapter is organized as follows. Theoretical investigation of SMC under high-dimensional systems is first provided. Then empirical studies on the effect of dimensionality on computation load are explored.

6.1 SMC and Curse of Dimensionality

As discussed in Section 3.3.3, theoretical studies indicate that the convergence of SMC in terms of mean-squares error toward zero is almost sure under weak assumptions [7]. Under these slightly restrictive conditions, the convergence rate is inversely proportional to the number of samples, and is independent of the dimensionality of the problem. Even though more studies are needed to explore the existence of a uniform convergence property, the number of samples required for SMC in high-dimensional systems may not have to be exponentially large. Thus the computation load of SMC may not increase dramatically with increasing dimensions.
Figure 6.1: Accurate estimation by Monte Carlo sampling may only need limited number of samples that well represent the key features of the posterior (as illustrated in colored boxes in (a) and (b)), thus may not need a set of samples which cover the whole solution space (as illustrated in (c) and (d)).

An intuitive way of explaining why SMC may not suffer much from the “curse of dimensionality” is that SMC may not need to have a set of samples that fill-up the whole solution space. In stead, only limited number of samples that well support the underlying posterior, may be required for SMC to have accurate estimation. Figure 6.1 provides a heuristic illustration of why the number of samples for accurate estimation may not need to grow exponentially according to the dimensionality. The colored area in Figure 6.1.(a) and 6.1.(b) implies the key features of the true posterior distribution. As long as these areas are represented by samples, there is no need to fill-up the whole as shown in 6.1.(c) and 6.1.(d).
Another promising property of SMC is that when samples are drawn from the true distribution, the covariance of the estimate error is almost inversely proportion to the number of samples [4], as discussed in Section 3.3.3.

The claim on the convergence rate to be independent from the dimensionality may not be easily accepted without further studies though. Indeed, SMC may require a strong assumption of being able to quickly “forget” about its previous estimation error to have uniform convergence [7]. Clearly, more research on both theoretical and empirical is needed to have a better understanding of this issue. This dissertation approaches this problem by empirically observing the effect on computation load according to the increment of dimensionality. In the next section, some practical solutions to reduce the effect of curse of dimensionality is suggested.

6.1.1 Practical Solutions

One benefit of SMC is that it can be readily implemented in parallel. This is because each sample can be computed independently for most of the computation algorithm. Parts of the algorithm that can not be incorporated into the parallel computation include the occasional implementation of resampling, and the inference from the posterior. Thus, a parallel implementation can retain the computational efficiency even when large number of samples become necessary.

Various practical methods exist to enhance the robustness of SMC. These variations of SMC usually improve the quality of importance functions by incorporating the information in the current measurement. For example, an optimal importance function may be available when the noise are additive Gaussian and the measurement equation is linear with additive Gaussian noise [11]. This optimal importance
function may be extended to nonlinear measurement equations by linearization. The algorithm in MCMC has also been suggested to be used in improving the quality of the importance function [3]. A hybrid approach that uses gradient decent information with SMC is also a possible venue [9]. Finally, EBSIR proposed in Section 3.4.4 is also expected to improve the quality of the importance function.

6.2 Illustrations and Case Studies

An eight dimensional (8-D) polymerization process is studied in this dissertation. This polymerization process of styrene is initialized by azo-bis-iso-butyro-nitrile (AIBN) with benzene as the solvent in a continuously stirred tank reactor (CSTR). Process model and simulation parameters can be found in [40], [17], and [38]. In order to observe the effect of dimensionality on computation load, two, five, and eight variables of this 8-D system are simulated and rectified separately, but examined the CPU requirement together. The simulation results indicate that the CPU load of SMC does not have to expand exponentially along with the increment of dimensionality.

In the following sections, the governing process model and simulation conditions are introduced first. Then two, five, and eight variables of the 8-D system are studied with three estimation methods, including EKF, MHE, and SMC. Finally, a discussion of the computation load in respect to the dimensionality is provided.

6.2.1 Process Model and Simulation Conditions

The governing process equations are provided as follows.

\[ \frac{dC_i}{dt} = -\frac{1}{\tau} C_i + f_1 , \]
\[ \frac{dC_s}{dt} = -\frac{C_s}{\tau} + f_2 , \]
\[
\begin{align*}
\frac{d\lambda_0}{dt} &= -\frac{\lambda_0}{\tau} + f_3(C_i, C_s, C_m, T), \\
\frac{d\lambda_1}{dt} &= -\frac{\lambda_1}{\tau} + f_4(C_i, C_s, C_m, T), \\
\frac{d\lambda_2}{dt} &= -\frac{\lambda_2}{\tau} + f_5(C_i, C_s, C_m, T), \\
\frac{dC_m}{dt} &= f_6(C_i, C_m, T), \\
\frac{dT}{dt} &= f_7(C_i, C_m, T, T_j), \\
\frac{dT_j}{dt} &= f_8(T, T_j),
\end{align*}
\]

where,

\[
\begin{align*}
f_1 &= \frac{Q_i C_i}{V}, \\
f_2 &= \frac{Q_s C_s + Q_i C_{si}}{V}, \\
f_3(C_i, C_s, C_m, T) &= [(k_{f_m} C_m + k_{t_d} P + k_{f_s} C_s)]\alpha P + \frac{1}{2}k_{t_c} P^2, \\
f_4(C_i, C_s, C_m, T) &= [(k_{f_m} C_m + k_{t_d} P + k_{f_s} C_s)(2\alpha - \alpha^2) \\
&\quad + k_{t_c} P] \frac{P M_m}{1 - \alpha}, \\
f_5(C_i, C_s, C_m, T) &= [(k_{f_m} C_m + k_{t_d} P + k_{f_s} C_s)(\alpha^3 - 3\alpha^2 \\
&\quad + 4\alpha) + k_{t_c} P(\alpha + 2)] \frac{P M_m^2}{(1 - \alpha)^2}, \\
f_6(C_i, C_m, T) &= \frac{Q_m C_{mm} - Q_t C_m - k_p C_m P}{V}, \\
f_7(C_i, C_m, T, T_j) &= \gamma k_p C_m P + \frac{U A(T_j - T)}{\rho C_p V} + \frac{Q_i(T_j n - T)}{V}, \\
f_8(T, T_j) &= \frac{U A(T - T_j)}{\rho_j C_p V_j} + \frac{Q_t(T_{jn} - T)}{V_j}, \\
\tau &= \frac{V}{Q_t}, \\
\alpha &= \frac{k_p C_m}{(k_p + k_{f_m}) C_m + k_{f_s} C_s + k_{t_c} P}, \\
P &= \sqrt{\frac{2f^* C_i k_i}{k_t}},
\end{align*}
\]
\[ \gamma = \frac{-\Delta H}{\rho C_p} \]

where \( C_i, C_s, C_m \) are the concentrations of the initiator, solvent and the monomer. Variables \( T \), and \( T_j \) are the temperatures in the CSTR reactor, and in the cooling jacket. First three moments of the molecular weight distribution (MWD) of the polymer are represented by \( \lambda_0, \lambda_1, \lambda_2 \). These eight variables are the state variables to be estimated in this dissertation. Definition of the rest of the variables can be found in Section 6.4.

The eight state variables at time \( k \) are augmented as \( x_k \), and defines as the following equation,

\[ x_k = \left[ \begin{array}{c} C_i; \frac{C_i}{C_i-r}; \frac{\lambda_0}{\lambda_1}; \frac{\lambda_1}{\lambda_2}; \frac{C_m}{C_m-r}; \frac{T}{T-r}; \frac{T_j}{T-r} \end{array} \right]^T_k, \]

where the denominators are the normalization constants which better conditioned the variables for simulation (Table 6.1). The continuous differential equations are discretized by finite difference with \( \Delta T = 1.0 \), and then added with system noise to form the system equations for simulation.

\[ x_{k+1} = \left[ \begin{array}{c} C_{i,k} + \Delta T \left( - \frac{1}{7} + k \right) C_{i,k} + f_1 \\ C_{s,k} + \Delta T \left( - \frac{C_s}{r} + f_2 \right) \\ \lambda_{0,k} + \Delta T \left( - \frac{\lambda_{0,k}}{r} + f_3(C_{i,k}, C_{s,k}, C_{m,k}, T_k) \right) \\ \lambda_{1,k} + \Delta T \left( - \frac{\lambda_{1,k}}{r} + f_4(C_{i,k}, C_{s,k}, C_{m,k}, T_k) \right) \\ \lambda_{2,k} + \Delta T \left( - \frac{\lambda_{2,k}}{r} + f_5(C_{i,k}, C_{s,k}, C_{m,k}, T_k) \right) \\ C_{m,k} + \Delta T \left( f_6(C_{i,k}, C_{m,k}, T_k) \right) \\ T_k + \Delta T \left( f_7(C_{i,k}, C_{m,k}, T_k, T_j) \right) \\ T_{j,k} + \Delta T \left( f_8(T_k, T_{j,k}) \right) \end{array} \right] + \omega_k \]

where \( \omega_k \) is the system noise. Measurement equation has the same format as in Equation (2.2). Stochastic properties of noises, \( \omega_k \) and \( \nu_k \), can be found in Table 6.1. Simulation parameters and initial conditions are provided in Table 6.2 [40], and 6.3 [17, 38].
$\Delta T$ & 1.0 & Simulation time span \\
$\{\omega_k\}$ & $N(\mu = 0.0, \sigma = 5.0 \times 10^{-4})$ & System noise \\
$\{\nu_k\}$ & $N(\mu = 0.0, \sigma = 5.0 \times 10^{-2})$ & System noise \\
$p(x_0)$ & $N(\mu = x_0, \sigma = 5.0 \times 10^{-4})$ & True initial state \\
$p(\hat{x}_0)$ & $N(\mu = x_0, \sigma = 5.0 \times 10^{-4})$ & Initial guess \\
$C_{i,R}$ & 0.07 kmol · m\(^{-3}\) & Normalizing constant \\
$C_{s,R}$ & 5.5 kmol · m\(^{-3}\) & Normalizing constant \\
$\lambda_0$ & $10^{-4}$ kmol · m\(^{-3}\) & Normalizing constant \\
$\lambda_1$ & 2.5 kg · m\(^{-3}\) & Normalizing constant \\
$\lambda_2$ & $10^5$ kg\(^2\) kmol\(^{-1}\) · m\(^{-3}\) & Normalizing constant \\
$C_{m,R}$ & 3.5 kmol · m\(^{-3}\) & Normalizing constant \\
$T_R$ & 350 K & Normalizing constant \\

$x_0$ is given by the initial conditions of Table 6.2.

Table 6.1: Simulation parameters of polymerization process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{i}(0)$</td>
<td>0.05</td>
<td>kmol · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$C_{s}(0)$</td>
<td>5.0</td>
<td>kmol · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$\lambda_0(0)$</td>
<td>0.0</td>
<td>kmol · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$\lambda_1(0)$</td>
<td>0.0</td>
<td>kg · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$\lambda_2(0)$</td>
<td>0.0</td>
<td>kg(^2) kmol(^{-1}) · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$C_{m}(0)$</td>
<td>3.0</td>
<td>kmol · m(^{-3})</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$T(0)$</td>
<td>330</td>
<td>K</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$T_j(0)$</td>
<td>295</td>
<td>K</td>
<td>Initial condition</td>
</tr>
<tr>
<td>$T_{in}$</td>
<td>330</td>
<td>K</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$T_{j_{in}}$</td>
<td>295</td>
<td>K</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$C_{i_{in}}$</td>
<td>0.5888</td>
<td>kmol · m(^{-3})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$C_{m_{in}}$</td>
<td>8.698</td>
<td>kmol · m(^{-3})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$C_{i}$</td>
<td>7.1</td>
<td>kmol · m(^{-3})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$C_{s}$</td>
<td>7.5</td>
<td>kmol · m(^{-3})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>$1.55 \times 10^{-5}$</td>
<td>m(^3) · s(^{-1})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$Q_j$</td>
<td>$6.55 \times 10^{-5}$</td>
<td>m(^3) · s(^{-1})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$Q_m$</td>
<td>$5.25 \times 10^{-5}$</td>
<td>m(^3) · s(^{-1})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$Q_{s}$</td>
<td>$6.375 \times 10^{-5}$</td>
<td>m(^3) · s(^{-1})</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$Q_{t}$</td>
<td>$Q_i + Q_m + Q_s$</td>
<td>m(^3) · s(^{-1})</td>
<td>Operating condition</td>
</tr>
</tbody>
</table>

Table 6.2: Operating conditions and simulation parameters of polymerization process.
Table 6.3: Operating conditions and simulation parameters of polymerization process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f^*$</td>
<td>0.6</td>
<td></td>
<td>Operating condition</td>
</tr>
<tr>
<td>$A_i$</td>
<td>$5.95 \times 10^{13}$</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$E_i$</td>
<td>14897</td>
<td>K</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$A_p$</td>
<td>$1.06 \times 10^7$</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$E_p$</td>
<td>3557</td>
<td>K</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$A_t$</td>
<td>$1.25 \times 10^9$</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$E_t$</td>
<td>843</td>
<td>K</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$\Delta H$</td>
<td>$-16700$</td>
<td>kJ·kmol$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$\rho \cdot C_p$</td>
<td>360</td>
<td>kJ·m$^{-3}$·K$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$\rho_j \cdot C_{pj}$</td>
<td>966.3</td>
<td>kJ·m$^{-3}$·K$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$UA$</td>
<td>0.007</td>
<td>kJ·s$^{-1}$·K$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$V$</td>
<td>3.0</td>
<td>m$^3$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$V_j$</td>
<td>3.33124</td>
<td>m$^3$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$k_{tc}$</td>
<td>$(1/9.23) \cdot k_t$</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$k_{td}$</td>
<td>$(8.23/9.23) \cdot k_t$</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$k_{fm}$</td>
<td>0.043</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$k_{fs}$</td>
<td>0.091</td>
<td>m$^3$·kmol$^{-1}$·s$^{-1}$</td>
<td>Operating condition</td>
</tr>
<tr>
<td>$M_m$</td>
<td>100.11</td>
<td></td>
<td>Operating condition</td>
</tr>
</tbody>
</table>
Comparison on estimation performance is based on the mean-squares error (MSE) and CPU load of 100 realizations of simulation. Throughout this chapter, number of measurements, \( N_m = 600 \), is used. In the above equation, \( x_{k,r} \) is the true state at time step \( k \) of the \( r \)-th realization, and \( \hat{x}_{k,r} \) is the point estimate of \( x_{k,r} \). Therefore, the values of MSE is based on the normalized state variables. In this dissertation, the posterior mean is chosen as the point estimate for SMC. Unfortunately, MSE often does not bring out the fact that SMC can provide much more information than MHE or EKF due to its more accurate posterior. Consequently, as demonstrated by the MSE comparisons in this section, the MSE for SMC may be comparable to that of MHE even for highly non-Gaussian posteriors. Research on using a more appropriate loss function for non-Gaussian and particularly, multi-modal distributions is in progress.

Simulation is run on a 3.4 GHz CPU with 1.0GB RAM personal computer using GNU/Octave, an open source implementation of Matlab. MHE is implemented by a publicly available code [15], which is considered to be robust and computationally efficient. Further improvement on the computation efficiency of MHE may not be significant since a customized optimization algorithm, SQP, is used and most parts of the code have already been compiled into fast binary format. Unlike MHE, SMC is implemented by a batch code, and further improvement on computation efficiency is expected.

6.2.2 Cases Studied

Three cases of the 8-D system are studied to examine the increment of computation load according to the dimensionality. The number of variables simulated are two, five
and eight which are chosen only for convenience. In the 2-D problem, $C_m$ and $T$ are chosen since concentration and temperature are two commonly studied variables in engineering problems. First three moments of the MWD of polystyrene, $\lambda_0$, $\lambda_1$, and $\lambda_2$, are ignored in the 5-D case, because the other five variables are dependent on each other, and is independent of these three neglected variables. Finally, in the 8-D case, all eight variables are used.

**Two Dimensional Problem.** State variables $C_m$ and $T$ are simulated with the assumption that all other states are static and available. Estimation result of $T$ of one realization is provided in Figure 6.2. All estimation methods provide similar estimation accuracy under this simulation condition. Boundaries of 95% of high probability density regions can be easily estimated by SMC, and illustrated as the
Table 6.4: Mean-squares error and CPU load of 2-D polymerization process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$C_m$</th>
<th>$T$</th>
<th>CPU time</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>$2.1 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.1 \times 10^{-5}$</td>
<td>0.0006 ± 0.0</td>
<td></td>
</tr>
<tr>
<td>MHE</td>
<td>$2.1 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.1 \times 10^{-5}$</td>
<td>$0.15 \pm 0.001$</td>
<td>$m = 5$</td>
</tr>
<tr>
<td></td>
<td>$2.1 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.1 \times 10^{-5}$</td>
<td>$0.21 \pm 0.003$</td>
<td>$m = 10$</td>
</tr>
<tr>
<td></td>
<td>$2.1 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.1 \times 10^{-5}$</td>
<td>$0.32 \pm 0.003$</td>
<td>$m = 20$</td>
</tr>
<tr>
<td>SMC</td>
<td>$2.1 \times 10^{-5} \pm 1.2 \times 10^{-5}$</td>
<td>$2.2 \times 10^{-5} \pm 1.2 \times 10^{-5}$</td>
<td>$0.009 \pm 0.0001$</td>
<td>$N = 500$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.2 \times 10^{-5}$</td>
<td>$0.02 \pm 0.0001$</td>
<td>$N = 1000$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 1.1 \times 10^{-5}$</td>
<td>$0.07 \pm 0.0002$</td>
<td>$N = 2000$</td>
</tr>
</tbody>
</table>

error bar. The confidence level of SMC was small in the beginning because the chosen initial guess has narrow covariance. Estimation performance is provided in Table 6.4 and Figure 6.3. Under this simulation condition, all methods have similar values of MSE. As discussed earlier, MSE may not be an ideal merit for representing the estimation quality of the posterior, especially in the multi-modal situation. SMC has been illustrated in the previous chapters to have superior performance when the posterior exhibits non-Gaussian features.

**Five Dimensional Problem.** State variables $C_i$, $C_s$, $C_m$, $T$, and $T_j$ are simulated and rectified in this case study. Again, the estimation result of $T$ of one realization is provided in Figure 6.4, and the confidence level of the posterior can be readily estimated by SMC, and illustrated by the error bar. Estimation performance
Figure 6.3: Estimation performance of 2-D polymerization process.

Figure 6.4: Rectification result of the temperature of 5-D polymerization process.
<table>
<thead>
<tr>
<th>$C_i$</th>
<th>$C_s$</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>EKF</td>
<td>$2.1 \times 10^{-9} \pm 1.3 \times 10^{-9}$</td>
<td>$2.5 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.1 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.5 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.1 \times 10^{-5} \pm 1.2 \times 10^{-5}$</td>
<td>$2.5 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
</tr>
<tr>
<td>MHE</td>
<td>$2.2 \times 10^{-9} \pm 1.4 \times 10^{-9}$</td>
<td>$2.6 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$2.6 \times 10^{-5} \pm 1.5 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.3 \times 10^{-5}$</td>
<td>$2.5 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
</tr>
<tr>
<td>$C_m$</td>
<td>$T$</td>
<td>Parameter</td>
</tr>
<tr>
<td>$Y$</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>EKF</td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$4.4 \times 10^{-5} \pm 2.9 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 2.1 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 2.1 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 2.1 \times 10^{-5}$</td>
</tr>
<tr>
<td>MHE</td>
<td>$2.3 \times 10^{-9} \pm 1.5 \times 10^{-9}$</td>
<td>$2.2 \times 10^{-5} \pm 2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.3 \times 10^{-5} \pm 1.5 \times 10^{-5}$</td>
<td>$2.2 \times 10^{-5} \pm 2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5} \pm 2.1 \times 10^{-5}$</td>
</tr>
<tr>
<td>$T_j$</td>
<td>CPU time</td>
<td>Parameter</td>
</tr>
<tr>
<td>$Y$</td>
<td>0.003</td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>$2.4 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$0.0009 \pm 0.0$</td>
</tr>
<tr>
<td></td>
<td>$2.4 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$0.20 \pm 0.002$</td>
</tr>
<tr>
<td></td>
<td>$2.4 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$0.30 \pm 0.003$</td>
</tr>
<tr>
<td></td>
<td>$2.4 \times 10^{-5} \pm 1.4 \times 10^{-5}$</td>
<td>$0.41 \pm 0.004$</td>
</tr>
<tr>
<td>MHE</td>
<td>$2.5 \times 10^{-5} \pm 1.5 \times 10^{-5}$</td>
<td>$0.01 \pm 0.00006$</td>
</tr>
<tr>
<td></td>
<td>$2.5 \times 10^{-5} \pm 1.5 \times 10^{-5}$</td>
<td>$0.03 \pm 0.0001$</td>
</tr>
<tr>
<td></td>
<td>$2.4 \times 10^{-5} \pm 1.5 \times 10^{-5}$</td>
<td>$0.10 \pm 0.0007$</td>
</tr>
</tbody>
</table>

Table 6.5: Mean-squares error and CPU load of 5-D polymerization process.
is provided in Table 6.5 and Figure 6.5. Similar to the 2-D case, all methods have similar estimation accuracy. Again, MHE has the heaviest computation load, and EKF is the fastest estimation method.

**Eight Dimensional Problem.** All eight variables are simulated and rectified in this case. Like previous two cases, the estimation result of $T$ of one realization is provided in Figure 6.6. Estimation performance is provided in Table 6.6 and Figure 6.7. Under this simulation condition, all methods have similar estimation accuracy. SMC has slightly inferior MSE than other approaches, which may result from the choice of using mean as the optimal estimate of the posterior. MHE systematically costs more computation power than SMC and EKF.
Figure 6.6: Rectification result of the temperature of 8-D polymerization process.

Figure 6.7: Estimation performance of 8-D polymerization process.
Table 6.6: Mean-squares error and CPU load of 8-D polymerization process.
6.2.3 Effect of Dimension on Computation Load

The simulation results indicate that SMC does not have to increase the computation load exponentially along with the increment of dimensionally. The increment of CPU time according to the dimensionality shows a trend less than the order of $O(N \cdot D)$, which $D$ is the ratio between the dimensionality. Increasing the number of samples CPU has more effect on the consumed computation power, than the one by increasing the dimensionality of the problem. The rate of increment seems to be slight larger than the ratio between the number of samples used.

The increment of computation load of MHE seems to have similar trend as of SMC’s that the degree of increment is less than $O(N \cdot D)$. However, MHE constantly demands much more than ten times of CPU consumption than SMC, and the ratio
of CPU load between MHE and SMC may be enlarged when SMC is implemented by compiled code.

6.3 Summary

In this chapter, effect of the dimensionality on the computation requirement of SMC has has also be explored with an application of SMC to a eight state-variable polymerization process. This is an empirical examination of the claim that SMC may be free from the “curse of dimensionality” [7]. Three different dimensions of the same polymerization process is studied, including 2-D, 5-D, and 8-D. Results show that the CPU load does increase to compensate the increased complexity, but in a rate implies that SMC may not explode in computation even when in high-dimensional systems. Even if SMC might require much more samples for accurate estimation, it is highly likely that SMC can work with the help of dimension reduction or parallel computation. Results of these studies also implies that online application of SMC may be feasible for large-scale systems.

6.4 Notation

\[A = \text{heat transfer area}\]
\[C_p = \text{heat capacity of the mixture in CSTR}\]
\[C_{p_j} = \text{heat capacity of the cooling fluid in the cooling jacket}\]
\[C_{i_i} = \text{concentration of the initiator in the initiator stream}\]
\[C_{i_i} = \text{concentration of the monomer in the monomer stream}\]
\[C_{s_i} = \text{concentration of the solvent in the solvent stream}\]
\[ C_{s,*} = \text{concentration of the solvent in the initiator stream} \]

\[ f^* = \text{initiator efficiency} \]

\[ k_{fm} = \text{rate for chain-transfer-to-monomer} \]

\[ k_{fs} = \text{rate for chain-transfer-to-solvent} \]

\[ k_i = A_i e^{-E_i/T} \text{dissociation rate for initiator} \]

\[ k_p = A_p e^{-E_p/T} \text{rate for propagation} \]

\[ k_t = A_t e^{-E_t/T} \text{rate for termination} \]

\[ k_{tc} = \text{rate for termination by combination} \]

\[ k_{td} = \text{rate for termination by disproportionation} \]

\[ M_m = \text{molecular weight of the monomer} \]

\[ P = \text{molar concentration of polymers with radical} \]

\[ Q_i = \text{inlet flow rate for initiator stream} \]

\[ Q_m = \text{inlet flow rate for monomer stream} \]

\[ Q_s = \text{inlet flow rate for solvent stream} \]

\[ Q_t = Q_i + Q_m + Q_s \]

\[ U = \text{heat transfer coefficient} \]

\[ \Delta H = \text{heat of propagation reaction} \]

\[ \rho = \text{density of the mixture in CSTR} \]

\[ \rho_j = \text{density of the cooling fluid in the cooling jacket} \]
CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This dissertation introduces a novel approach for estimation or rectification in nonlinear dynamic process systems. This approach is based on a rigorous Bayesian formulation of estimation problems that uses sequential Monte Carlo (SMC) sampling to propagate all information recursively, while minimizing assumptions about the system. The resulting approach does not rely on common assumptions of Gaussian or fixed-shape distributions, which can be easily violated in nonlinear dynamic systems. Therefore this proposed approach is expected to handle broader range of problems and result in more accurate estimation than existing approaches. These expectations are supported by various cases studies in this dissertation. Although SMC has been successfully applied in various areas of research, it has not been introduced in process system engineering. In addition to applying the available algorithm, systematical comparison of SMC with existing methods is also provided. There is no such comparison in existing literatures. Furthermore, this dissertation proposes a novel method for dealing with the practical challenges faced by SMC methods. This
approach, called Empirical Bayes Sampling Importance Resampling (EBSIR), provides better accuracy with more convenient tuning parameters than existing SMC methods. Comparison of mean-squares error and CPU load are conducted with existing methods, including extended Kalman filtering (EKF), and moving horizon based least-squares estimation (MHE). As expected, SMC outperforms these existing methods when the underlying distributions exhibit non-Gaussian distribution, and has at least as accurate estimation as MHE when distributions are almost Gaussian.

Gaussian approximation does not necessarily provide computational benefits for estimation. MHE relies on Gaussian approximation so that a formulation of least-squares estimation (LSE) may be found. MHE may be more accurate than EKF by retaining nonlinearity and being able to handle constraints, but there is no closed-from solution for MHE anymore. Instead, MHE relied on computationally expensive algorithms like nonlinear or quadratic programming to solve constrained optimization problems within each horizon window. Furthermore, MHE lacks a full recursive formation since it is tailored for estimating the mode of the posterior only, and does not have an accurate and convenient way of propagating enough information of the posterior for next horizon. Therefore, MHE is a computationally demanding algorithm. In addition to the potentially inferior estimation and intensive demand of computation, MHE may suffer from the use of nonlinear or quadratic programming techniques by being trapped in an unlikely local minimum, as implied in the case study of a nonlinear dynamic system in Section 4.2.2. Convergence of MHE may be guaranteed, but the convergence rate may not be practical for estimation.

MHE may be able to handle non-Gaussian distributions by mixtures of Gaussian distributions, but such Gaussian sum method may not be practical for large-scale
systems. On the contrary, SMC handles non-Gaussian distribution naturally, and does not have to change the formulation. This advantage over MHE is illustrated in Section 5.2.2 when a non-Gaussian measurement noise exists. SMC follows the same algorithm suggested in Table 3.3, but MHE requires a modification on problem formulation, and the optimization solver need to be rewritten. Furthermore, Gaussian sum method may work with static non-Gaussian distributions, but may not be practical in approximating dynamic evolution of posteriors.

SMC handles process constraints naturally with minor modification of formulation. In this dissertation, an acceptance/rejection algorithm that keeps qualified samples, while rejects the rest is implemented to enforce process constraints. The benefit of handling constraints becomes evident as shown in the cases studied in Section 5.2. EKF is disadvantaged by not enforcing constraints, and results in poor estimates falling into regions where constraints are violated. Both SMC and MHE can handle constraints, but SMC can outperform MHE in estimation accuracy since the underlying distributions are usually truncated, and non-Gaussian.

Effect of the dimensionality on the computation requirement of SMC has also been explored in this dissertation. Recent theoretical studies expect SMC to free from the “curse of dimensionality” [7]. This dissertation aims to verify such claim through a study of the correlation between the computation load and the dimensionality empirically. An eight variable polymerization process was studied with three different number of dimensionality, including 2-D, 5-D, and 8-D. Results show that the CPU load does grow with higher dimensionality to compensate the increased complexity, but indicates that the increment in CPU cycles may not be significant
even for higher-dimensional problems. This implies that online application of SMC on large-scale systems may be feasible.

The ease of finding the confidence level demonstrates one possible use of the posterior estimate of SMC. Comparing to methods tailored for providing point estimate, SMC has the whole posterior as the estimate, and various moments of the posterior may be accurate determined. As mentioned earlier, MHE lacks a convenient algorithm of finding the covariance, thus fails to have a full recursive formulation for its LSE formulation. Other potential use of the whole posterior as the estimate includes tracking multiple modes of the posterior for control purpose, and observing the transition of dominate states.

In principle, Bayesian estimation by sequential Monte Carlo sampling can handle a wide range of practical situations in dynamic estimation problems, including linear or nonlinear dynamics, Gaussian or non-Gaussian distributions, bias, constraints, and missing or multi-rate data. Although this dissertation has only focused on data rectification of nonlinear dynamic systems, the approach is broadly applicable to most process engineering tasks. The overall improvement on industrial process by applying SMC is expected to permit a higher return on the investment in corporate networks and process knowledge.

7.2 Future Work

Areas where SMC may potentially have significant impact are in those where data usually exhibit non-Gaussian features. One immediate area of exploration is in bias/outliers detection where the bias/outliers are usually approximated with non-Gaussian distributions of long tails. A popular way of representing such non-Gaussian
distributions is to approximate the distributions with mixtures of two Gaussian distributions, and pronounce a noise is a bias/outliers when the chance of that noise is from one of the Gaussian distributions with larger covariance. SMC is also expected to have similar on accuracy and computational efficiency is dissertation in multi-rate/missing data problems. Multi-rate problems are common in engineering practices that measurements may be sampled with different sampling period. In fact, the polymerization process studied in Chapter 6 was originally studied for multi-rate estimation.

In this dissertation, SMC has shown promising results for its application in large-scale problems. The CPU load of SMC increases slightly along with the increment of dimensionality. This result may indicate that SMC does not need to dramatically increase the number of samples for accurate estimation in even higher dimensional problems. Therefore, the required computation load may be in an acceptable scale for online application of SMC in large-scale problems. In addition to explore the performance of SMC in even larger systems, SMC is also likely to benefit from parallel computation or simply using compiled code. Parallel computation of SMC is expected to have significant reduction in overall computation period, since many parts of the algorithm of SMC may be processed independently by different computers. These steps involve sampling from the importance function, calculating the likelihood values, and updating the weight. Like parallel computation, binary code is likely to be able to use larger number of samples in a similar time period compared to an implementation by batch code. Therefore, larger number of samples may be used for estimation, and this usually results in more accurate estimation, and may be more stable. Combination of principal component analysis (PCA) and SMC may also have
substantial impact on process engineering tasks, since most of measured samples may be dependent on each other.

Further studies on the requirements of the convergence properties of SMC, including the prediction of the minimum number of samples required, and a general and practical formulation of SMC with uniform convergence, are expected to have both theoretical and methodological benefits. In addition to theoretical studies, SMC is more likely to encounter convergence issue when there is little overlap between the prior and likelihood distributions, and may use this phenomenon as an indicator for further investment of efforts. In case of such events, adopting another importance function with better chance of faster convergence may alleviate the potential slow convergence phenomenon. Candidate importance functions include the use of Markov Chain Monte Carlo (MCMC, [1]), the “optimal” importance function [11], the Gaussian particle filtering [22], or the EBSIR approach suggested in this research. In case when These “smarter” importance functions demand more computation than the convenient importance function, \( p(x_k|x_{k-1}(i)) \), a hybrid approach capable of switching the use of importance functions may in average provide a decent balance between efficiency and accuracy.

Many more areas are worth pursuing with SMC. Relaxation on the assumptions in this dissertation that perfect information of process model and statistical properties of noise are available, may pave the path for more realistic application of SMC. Combination of multi-scale analysis and SMC may be able to integrate the benefits from either formulation. A more appropriate metric that better illustrates the benefit of SMC which has the whole posterior as the estimate may provide more insights of the potential of SMC.
APPENDIX A

SOURCE CODE

The source code of SMC is provided and briefly explained in this section. This code should work both in Mathworks/Matlab and GNU/Octave. For convenience, the code for nonlinear dynamic system in Section 4.2.2 is used.

The core file of implementing SMC, `smc_gordon_nmhe.m`, is provided in the next paragraph. For each time step $k$, SMC performs three main tasks, including drawing samples from the importance function, calling `smc_pred_gordon_nmhe.m`, updating the weight, calling `smc_lh.m`, and deciding if resampling is necessary, calling `smc_rs.m`. Figure A.1 depicts how SMC is implemented at each time step.

```matlab
function smcX=smc_gordon_nmhe(Y,smcXp)
% smcX estimated X
% smcXp samples from the initial guess
% Y noisy measurements

global svN N_m N N_threshold R
% svN number of state variables
% N_m temporal length of measurements
% N number of samples for SMC
```

104
% N_threshold pre-determined threshold for resampling
% R        error covariance of the measurement noise
smcX=zeros(svN,N_m);
smcWp=1/N*ones(1,N);
for k=1:N_m, % k=1 is the initial state, or initial guess
  %k
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Stage 1. Drawing samples from the importance function
  smcXm=smc_pred_gordon_nmhe(smcXp,k);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Stage 2. Updating the weight
  e=Y(:,k)*ones(1,N)-(smcXm.^2)./20;
  % Find the likelihood value
  [smcWm]=smc_lh(e,R);
  smcWp=(smcWp.*smcWm)./sum(smcWp.*smcWm);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Stage 3. deciding if resampling is necessary
  % Estimating Effective Sample Size
  N_ess=inv(sum(smcWp.^2));
  % resampling when necessary
  if N_ess < N_threshold,
    % Resampling is implemented
    [smcXp,randRS(:,k)]=smc_rs(smcXm,smcWp); % resampling
    smcWp=1/N*ones(1,N);
  end
end
else
    % Resampling is not necessary
    smcXp=smcXm;
end
for varI=1:svN,
    smcX(varI,1)=sum(smcWp.*smcXp(varI,:));
end

The first stage of SMC is to draw samples from the importance function. M-file
smc_pred_gordon_nmhe.m handles the generation of samples from the importance
function \( p(x_k|x_{k-1}(i)) \), and is provided as follows.

function smcXm=smc_pred_gordon_nmhe(smcXp,k)
    % Model prediction stage of SMC
    global svN Q N
    % R error covariance of the system noise
    EstSysNoise=sqrt(Q)*randn(svN,N);
    smcXm=(0.5.*smcXp)+((25.*smcXp)./(1+(smcXp.^2)))+...
        8*cos(1.2*(k-1))+EstSysNoise;

The output smcXm comprise of samples drawn from the importance function \( p(x_k|x_{k-1}(i)) \).
The \( i \)-th sample of the posterior at time step \( k - 1 \), \( x_{k-1}(i) \), is propagated from
smcXp(i), and the \( i \)-th sample of the prior is stored in smcXm(i).

The second stage of SMC is to update the weight. M-file smc_lh.m calculates the
likelihood values of \( p(y_k|\text{smcXm}) \), representing the weights based on current measurement only. The value of \( p(y_k|\text{smcXm}) \) can be easily found as \( C \cdot \exp\left(-\frac{1}{2} e' R^{-1} e\right) \), since
Inputs: Y, smcXp, smcWp

Figure A.1: Flow chart of SMC code.
the measurement noise is additive Gaussian, where $C$ is the normalizing constant, and
\[ e = y_k - \frac{smcXm^2}{20}, \]
captures the difference between the current measurement and the predicted measurement value by the available sample.

function \[smcW]=smc_lh(e,R)\]
% update and normalize the weight
smcLH=exp(-0.5.*diag(e'*inv(R)*e))';
smcW=smcLH./sum(smcLH);

The updated weight is the normalized values of the multiplication of previous weight and current likelihood value.

The last stage of SMC is to check if resampling is necessary. If resampling is required, \texttt{smc_rs.m} is called.

function smcXp=smc_rs(smcXm,smcW)
% implementing resampling
global N
tmp_smcW=cumsum(smcW);
uniform_random_pick=rand(N,1);
smcRSI=zeros(1,N); % index for "picked" sample
for rsI=1:N,
    smcRSI(rsI)=min(find(uniform_random_pick(rsI)-tmp_smcW<0));
end
smcXp=smcXm(:,smcRSI);

As illustrated in Figure 3.4, the cumulative sum of normalized weights is calculated first, and a random number between 0 and 1 is generated. Based on the region where
the random number falls in, the corresponding sample is considered to be resampled for representing the posterior.
APPENDIX B

DERIVATION OF EXISTING METHODS VIA BAYESIAN ESTIMATION

B.1 Derivation of Kalman Filtering via Bayesian Estimation

At time step \( k \), Kalman filtering assumes the prior is available and Gaussian. If the mean of the prior is \( \mu_{x_{k-1}} \) and the covariance is \( Q_{x_{k-1}} \), the probability function of the prior may be represented as the following equation.

\[
p(x) \propto e^{-\frac{1}{2}(x_{k-1} - \mu_{x_{k-1}})^T Q_{x_{k-1}}^{-1} (x_{k-1} - \mu_{x_{k-1}})}.
\]

The model prediction of Kalman filtering involves solving Equations (2.8) and (2.9). Equation (2.9) may be first manipulated as follows.

\[
p(x_k| x_{k-1}) \propto \int \delta(x_k - F_{k-1} x_{k-1} - \omega_{k-1}) e^{-\frac{1}{2} \omega_{k-1}^T Q_{\omega_{k-1}}^{-1} \omega_{k-1}} d\omega_{k-1},
\]

\[
\propto e^{-\frac{1}{2}(x_k - F_{k-1} x_{k-1})^T Q_{k-1}^{-1} (x_k - F_{k-1} x_{k-1})},
\]

\[
\log[p(x_k|x_{k-1})] \propto -(x_k - F_{k-1} x_{k-1})^T Q_{\omega_{k-1}}^{-1} (x_k - F_{k-1} x_{k-1}),
\]

where \( Q_{\omega_{k-1}} \) is the known covariance of system noise, \( \omega_{k-1} \). Inserting this result into Equation (2.8) completes the prediction stage of KF.

\[
p(x_k|y_{1:k-1}) \propto e^{-\frac{1}{2}(x_k - F_{k-1} x_{k-1})^T Q_{\omega_{k-1}}^{-1} (x_k - F_{k-1} x_{k-1})} \times
\]

\[
e^{-\frac{1}{2}(x_{k-1} - \mu_{x_{k-1}})^T Q_{x_{k-1}}^{-1} (x_{k-1} - \mu_{x_{k-1}})},
\]

110
\[
\log[p(x_k|y_{1:k-1})] \propto -\{(x_k - F_{k-1} x_{k-1})^T Q_{\omega k-1}^{-1} (x_k - F_{k-1} x_{k-1}) + \\
(x_{k-1} - \mu_{x_{k-1}})^T Q_{x_{k-1}}^{-1} (x_{k-1} - \mu_{x_{k-1}})\},
\]

For systems with process model as Equations (2.11) and (2.12), \( p(x_k|y_{1:k-1}) \) is Gaussian with mean \( F_{k-1} \mu_{x_{k-1}} \) and covariance \( F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega k-1} \). That is,

\[
\log[p(x_k|y_{1:k-1})] \propto -\{(x_k - F_{k-1} \mu_{x_{k-1}})^T (F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega k-1})^{-1} (x_k - F_{k-1} \mu_{x_{k-1}})\}.
\]

Likelihood may be calculated as follows.

\[
p(y_k|x_k) \propto \int \delta(y_k - h_k(x_k, \nu_k)) e^{-\frac{1}{2} \nu_k^T Q_{\nu k}^{-1} \nu_k} d\nu_k,
\]

\[
\log(p(y_k|x_k)) \propto - (y_k - H_k x_k)^T Q_{\nu k}^{-1} (y_k - H_k x_k),
\]

where \( Q_{\nu k} \) is the known covariance of the measurement noise, \( \nu_k \).

The posterior log may be found by multiplying \( p(x_k|y_{1:k-1}) \) and \( p(y_k|x_k) \) as,

\[
\log[p(x_k|y_{1:k})] \propto -\{(y_k - H_k x_k)^T Q_{\nu k}^{-1} (y_k - H_k x_k) + \\
(x_k - F_{k-1} x_{k-1})^T Q_{\omega k-1}^{-1} (x_k - F_{k-1} x_{k-1}) + \\
(x_{k-1} - \mu_{x_{k-1}})^T Q_{x_{k-1}}^{-1} (x_{k-1} - \mu_{x_{k-1}})\},
\]

\[
\times -\{(y_k - H_k x_k)^T Q_{\nu k}^{-1} (y_k - H_k x_k) \\
+ (x_k - F_{k-1} \mu_{x_{k-1}})^T (F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega k-1})^{-1} \times \\
(x_k - F_{k-1} \mu_{x_{k-1}})\}.
\]

Expanding and rearranging the squares terms may result in the following equation [18].

\[
\log[p(x_k|y_{1:k})] \propto -\{(x_k - \mu_{x_k})^T Q_{x_k} (x_k - \mu_{x_k})\},
\]

111
where

\[ \mu_{x_k} = F_{k-1} \mu_{x_{k-1}} + K_k (y_k - H_k F_{k-1} \mu_{x_{k-1}}) , \]  
\[ Q_{x_k}^{-1} = (F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega_{k-1}})^{-1} + H^T Q_{v_k}^{-1} H , \]  
\[ K_k = (F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega_{k-1}}) H^T \times \]  
\[ (H (F_{k-1} Q_{x_{k-1}} F_{k-1}^T + Q_{\omega_{k-1}}) H^T + Q_{v_k})^{-1} . \]

Equations (B.1) to (B.3) are exactly the solution for KF.

B.2 Derivation of Moving Horizon Estimation via Bayesian Estimation

Derivation of MHE via Bayesian estimation is very similar to the derivation of KF. At time step \( k \), the prediction of state \( k - m + 1 \) is assumed to be available and Gaussian by MHE. \( m \) is the window width. The probability function may be represented as the following equation.

\[
p(x_{k-m+1}|y_{1:k-m}) \propto e^{-\frac{1}{2} (x_{k-m+1} - f_{k-m}(\mu_{x_{k-m}}))^T Q_{x_{k-m}}^{-1} (x_{k-m+1} - f_{k-m}(\mu_{x_{k-m}})) ,}
\]

\[
\log(p(x_{k-m}|y_{1:k-m})) \propto -\{(x_{k-m+1} - f_{k-m}(\mu_{x_{k-m}}))^T Q_{x_{k-m}}^{-1} \times \]  
\[ (x_{k-m+1} - f_{k-m}(\mu_{x_{k-m}})) \} ,
\]

where \( \mu_{x_{k-m}} \) is the mean, and \( Q_{x_{k-m}} \) is the covariance of the distribution.

For the \( i \)-th model prediction within the moving window,

\[
\log[p(x_{k-m+i+1}|x_{k-m+i})] \propto \]  
\[ -(x_{k-m+i+1} - f_{k-m+i}^*(x_{k-m+i}))^T Q_{\omega_{k-m+i}}^{-1} (x_{k-m+i+1} - f_{k-m+i}^*(x_{k-m+i})) .
\]

Likewise, the \( i \)-th likelihood within the moving window,

\[
\log[p(y_{k-m+i}|x_{k-m+i})] \propto
\]
\[-(y_{k-m+i} - h^*_{k-m+i}(x_{k-m+i}))^T Q^{-1}_{k-m+i} (y_{k-m+i} - h^*_{k-m+i}(x_{k-m+i}))\].

The posterior may be found as [34],

\[
\log[p(x_{k+m+1:k}|y_{1:k})] \propto -\left\{ \sum_{i=1}^{m} \nu_{k-m+i}^T Q^{-1}_{\nu} \nu_{k-m+i} + \sum_{i=1}^{m-1} \omega_{k-m+i}^T Q^{-1}_{\omega} \omega_{k-m+i} \right. \\
\left. + (x_{k-m+1} - f^*_{k-m}(\mu_{x_{k-m}}))^T Q^{-1}_{x_{k-m}} (x_{k-m+1} - f^*_{k-m}(\mu_{x_{k-m}})) \right\},
\]

where

\[
\omega_{k-m+i} = x_{k-m+i+1} - f^*_{k-m+i}(x_{k-m+i}) , \text{ and} \\
\nu_{k-m+i} = y_{k-m+i} - h^*_{k-m+i}(x_{k-m+i}).
\]


[5] Wen-shiang Chen and Bhavik R. Bakshi. A tutorial on importance sampling. Technical report, Department of Chemical Engineering, The Ohio State University, Columbus, OH.


