Sequential Monte Carlo Methods for Data Assimilation In Strongly Nonlinear Dynamics

Zhiyu Wang

B.Sc., Nanjing University, (China) 2005

Thesis Submitted In Partial Fulfillment Of

The Requirements For The Degree Of

Master of Natural Resources and Environmental Studies

The University of Northern British Columbia

April 2009

Your file Votre référence
ISBN: 978-0-494-48741-9
Our file Notre référence
ISBN: 978-0-494-48741-9

Abstract

General Bayesian estimation theory is reviewed in this study. The Bayesian estimation provides a general approach to handle nonlinear, non-Gaussian, as well as linear, Gaussian state estimation problems. The Sequential Monte Carlo (SMC) methods are presented to solve the nonlinear, non-Gaussian estimation problems. We compare the SMC methods with the Ensemble Kalman Filter (EnKF) method by performing data assimilation in the nonlinear, non-Gaussian dynamics. The Lorenz 1963 and 1996 models serve as test beds for examining the properties of these two estimation methods.

Although EnKF computes only mean and variance based on the assumption of Gaussian dynamics, the SMC methods do not outperform EnKF in practical applications of the nonlinear non-Gaussian cases as we expect in theoretical insights. The reasons behind the experimental results that the SMC methods perform as well as EnKF in data assimilation and the future applications for high dimensional realistic atmospheric and oceanic models are discussed.

To my beloved parents

Acknowledgements

First and foremost, it is difficult to overstate my gratitude to my supervisor, Dr. Youmin Tang. Without his enthusiasm, his inspiration, and his great efforts, I would have been lost.

I would also like to thank my graduate supervisory committee members Dr. Stephen J. Déry and Dr. Liang Chen for all the support they have given to me during the past years.

I am indebted to my colleagues Dr. Xiaobing Zhou, Dr. Ziwang Deng, Mr. Yanjie Cheng, Mr. Jaison T. Ambadan, and Ms. Xiaoqin Yan for providing a stimulating and fun environment in which to learn and grow. In particular I would like to acknowledge the help of Dr. Youqin (Jean) Wang for her support at HPC Lab, UNBC.

I am grateful to my professors, colleagues, and friends, Dr. Chris Johnson, Dr. Andrew Clifton, Dr. Jinjun Tong, and Mr. Joshua Laurin, from the University of Northern British Columbia and the City of Prince George for their continued support in these years.

Finally, I am forever thankful to my parents for their understanding, endless patience and encouragement when it was most required.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Data Assimilation: A Brief Review

What is data assimilation? In atmospheric and oceanic research, data assimilation is defined by Talagrand (1997) as the process to estimate the state of a dynamic system such as atmospheric and oceanic flow as accurately as possible by combining the observational and model forecast data.

From this perspective, a data assimilation system consists of three components: a time-evolving dynamic model, a measurement model for observations, and a data assimilation method. Dynamic models are not perfect due to sub grid physics parameterizations, physical process approximations, continuum fluid discretization into numerical scheme, etc. Similarly, instrument errors and representative errors cannot be avoided in a measurement model. Errors from both the dynamic model and measurement model add up to the essential concept that error plays a central and critical role in data assimilation; or rather error must be accurately estimated and modeled.

1.2 State-Space Form

In atmospheric and oceanic data assimilation, geophysical flow is usually described by a system of stochastic partial differential equations (sPDE). Within this framework, not only could the dynamic system be stochastically forced, but observations are also considered as stochastic processes rather than single numerical values. The most commonly used sPDE model is the nonlinear state-space model, which consists of a system of first order nonlinear differential equations. The dynamic model describes the evolution of the state variables over time, whereas the measurement model explains how the measurements relate to the state variables,

$$x_{t+1} = f(x_t, w_t, \theta, t) \qquad \text{(dynamic model)} \qquad (1.1)$$

$$y_t = h(x_t, e_t, \theta, t) \qquad \text{(measurement model)} \qquad (1.2)$$

where x denotes the state variable, θ denotes the time-invariant parameter, t denotes time, w_t and e_t denote stochastic forcings, commonly referred to as the dynamic process noise and the measurement noise. The functions f and h describe the evolution of the state variable and the measurements over time.

1.3 Existing Methods

Up to date, data assimilation in atmospheric sciences and oceanography can be divided into two categories: variational methods and sequential methods. Variational methods such as three-dimensional variational (3D-VAR) data assimilation and four-dimensional variational (4D-VAR) data assimilation (Dimet & Talagrand, 1986; Courtier *et al.*, 1998) relate to control theory framework, while sequential methods such as Kalman filter proposed by Kalman (1960) belong to estimation theory framework. They both have had great success. The European Centre for Medium-Range Weather Forecasts (ECMWF) introduced the first 4D-VAR methods into the operational global analysis system in the world in November 1997 (Rabier *et al.*, 2000; Mahfouf & Rabier, 2000; Klinker *et al.*, 2000). Ensemble Kalman Filter (EnKF) was first introduced into the operational ensemble prediction system in January 2005 by Canadian Meteorological Centre (CMC) (Houtekamer *et al.*, 2005).

Among sequential data assimilation methods, the most special case occurs when all equations are linear and the noise terms are Gaussian. The solution is in this case provided by the Kalman filter introduced by Kalman (1960). Furthermore, in the nonlinear case, approximate techniques have to be employed. A common idea is to linearise the nonlinear model, which results in the Extended Kalman Filter (EKF) (Smith *et al.*, 1962; Schmidt, 1966).

Another popular variety of Kalman filter is the Ensemble Kalman Filter (EnKF), initially introduced by Evensen (1994). It has a simple conceptual formulation and is easy to implement compared to other sophisticated assimilation methods such as 4D-VAR (Courtier *et al.*, 1998). Moreover, EnKF avoids many of the problems associated with the traditional EKF, for example, there is no closure problem as is introduced in the EKF by neglecting contributions from higher-order statistical moments in the error covariance evolution equation. There are numerous applications for EnKF (Evensen & van Leeuwen, 1996; Houtekamer & Mitchell, 1998; Burgers *et al.*, 1998; Tippett *et al.*, 2003; Evensen, 2003; Lorenc, 2003).

1.4 General Case: Nonlinear, Non-Gaussian

Although Kalman Filter type methods gained great success in applications of atmospheric and oceanic sciences, they are derived and validated for the linear dynamic system and Gaussian noise. Even in the well-known EnKF, there is an inherent assumption that the error statistics are Gaussian because only mean and covariance of data are employed to characterize the error. That may not be true for some nonlinear dynamics. In nonlinear dynamic systems, even though the initial error distribution is Gaussian, in general, it does not remain Gaussian with the forward evolution of the model.

In the Kalman filter framework, nonlinearity and non-Gaussianity problems cannot be solved theoretically. Therefore, to tackle this problem of nonlinear, non-Gaussian system estimation, the probability density function (PDF) associated with the dynamic system is used as a powerful tool to characterize the dynamic system uncertainty instead of only mean and covariance of the system data (Jazwinski, 1970). Statistics such as mean and variance can be calculated directly from the PDF. This class of methods keeps the original nonlinear dynamic model and tries to approximate the optimal solution, that is, the probability density function associated with the dynamics. In statistics, this class of methods is defined as Sequential Monte Carlo (SMC) methods, also known as particle filter, which is conceptually promising when the model is nonlinear. The first successful practical application of SMC methods is done by Gordon *et al.* (1993).

1.5 Research Objective

Although the Ensemble Kalman Filter (EnKF) method has been widely used in the data assimilation field and achieved great success, data assimilation problems in nonlinear, non-Gaussian dynamics still need to be solved. Sequential Monte Carlo (SMC) methods as a promising method show a great potential in solving nonlinear, non-Gaussian problems. In this thesis, we will investigate the performance and capability of SMC methods for data assimilation in highly nonlinear dynamics, and we will compare all the results from SMC methods with those from EnKF method in the same scenarios, finally we will discuss some drawbacks of SMC methods in realistic applications.

The atmospheric and oceanic flow has strongly nonlinear and chaotic nature. The dynamic models used in this thesis are the Lorenz 1963 and 1996 models. The Lorenz models are simplified atmospheric and oceanic models with the nature of realistic atmosphere and ocean. They can be used as test beds for data assimilation in atmospheric and oceanic fields. Although they are highly nonlinear dynamic models with stochastic characteristics, still they are relatively low dimensional models so that it is easier to perform the new data assimilation methods with them before they can be applied to high dimensional realistic atmospheric and oceanic models. Therefore, experiments with Lorenz models are computationally economical and realistically sufficient.

1.6 Outline of Thesis

The Sequential Monte Carlo (SMC) methods will be used within the probabilistic framework to tackle nonlinear and non-Gaussian estimation problems. SMC methods avoid deriving an inverse or an adjoint model and make them easier to adapt to all models. This thesis is concerned with the problem of estimating the state of variables in nonlinear dynamic systems. In the meantime, Ensemble Kalman Filter (EnKF) will also be used in the same scenarios for the sake of comparison.

Chapter1 introduces the idea of data assimilation, and the most widely used data assimilation methods, 3D-VAR, 4D-VAR and EnKF. To solve the nonlinear, non-Gaussian data assimilation problem, Sequential Monte Carlo (SMC) methods are developed.

Chapter2 gives a brief review of Kalman filter type data assimilation methods, Extended Kalman Filter (EKF) and Ensemble Kalman Filter (EnKF). Sequential Monte Carlo (SMC) methods are introduced, especially the implementation of SMC methods.

Chapter3 demonstrates the applications of the SMC methods and the EnKF method in the Lorenz 1963 model with different configurations of experiments.

Chpater4 further shows the applications of the SMC methods and the EnKF method in the Lorenz 1996 model with different chaotic degrees.

Chapter5 presents discussions and conclusions, and highlights the possible future research for Sequential Monte Carlo methods in data assimilation field for realistic dynamic models.

Appendix A and Appendix B give the detailed Fortran code to implement Ensemble Kalman Filter (EnKF) and Sequential Monte Carlo (SMC) methods.

Chapter 2

Sequential Data Assimilation Methods

2.1 Nonlinear State Estimation

Recursive nonlinear state estimation is addressed mainly within a probabilistic framework, that is, the Bayesian estimation theory (Cohn, 1997). In this framework, data assimilation, or rather state estimation, is simple enough to understand conceptually. We estimate the probability density function (PDF) for the current model state as accurately as possible given all the present and past observations. This implies that the complete solution to the estimation problem is provided by the conditional probability density function $p(x_t|Y_t)$. x_t denotes the state variable at time t, Y_t denotes all the observations up to time t and including time t. This conditional probability density function $p(x_t|Y_t)$ contains all the available information about the state variable.

Bayesian Recursive State Estimation If the dynamic model is given by (1.1) and the measurement model is given by (1.2), the target conditional probability density function to be estimated $p(x_t|Y_t)$, the one step ahead forecast probability density function $p(x_t|Y_{t-1})$ is given by

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \tag{2.1}$$

$$p(x_t|Y_{t-1}) = \int\limits_{\mathbf{R}^{nx}} p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1})dx_{t-1} \tag{2.2}$$

where

$$p(y_t|Y_{t-1}) = \int\limits_{\mathbf{R}^{nx}} p(y_t|x_t)p(x_t|Y_{t-1})dx_t \tag{2.3}$$

From (2.1), to obtain the conditional probability density function $p(x_t|Y_t)$, we need the observational noise probability density function $p(y_t|x_t)$, one step ahead forecast probability density function $p(x_t|Y_{t-1})$ which is the prior knowledge of the state variable, and marginal observational probability density function $p(y_t|Y_{t-1})$. Since the probability density function $p(y_t|x_t)$ can be calculated from the measurement model, the one step ahead forecast probability density function $p(x_t|Y_{t-1})$ can be calculated from the dynamic model, and $p(y_t|Y_{t-1})$ can be calculated according to (2.3). The target probability density function $p(x_t|Y_t)$ can be estimated. After that, with the new observation coming in and the dynamic model forward evolution, this estimation algorithm becomes recursive (Doucet *et al.*, 2001; Schon, 2006).

However, in general, there is no analytical solution to the nonlinear recursive estimation problem. This implies that we are forced to make approximations to approach this problem. The approximations suggested in the literature so far, can roughly be divided into two different classes, local approach and global approach (Schon, 2006). It is a matter of either approximating the nonlinear model and using the linear, Gaussian model estimator such as Extended Kalman Filter (EKF) or using the original nonlinear model and approximating the optimal solution such as Sequential Monte Carlo (SMC) methods. Despite the fact that there is a lot of different nonlinear estimators available, the local approximation approach is still the most commonly used nonlinear estimator when it comes to practical applications (Smith *et al.*, 1962; Schmidt, 1966; Evensen & van Leeuwen, 1996; Houtekamer & Mitchell, 1998; Burgers *et al.*, 1998; Tippett *et al.*, 2003; Evensen, 2003; Lorenc, 2003).

Local Approach The idea employed in local methods is to approximate the nonlinear model by a linear, Gaussian model, which is called the linearization process. This linearized model is only valid locally, but the Kalman Filter (Kalman, 1960) can directly be applied. This is the principle of the Extended Kalman Filter (EKF) (Smith *et al.*, 1962; Schmidt, 1966). For a more thorough treatment of the EKF, please refer to Jazwinski (1970) and Anderson & Moore (1979).

Global Approach The solution to the nonlinear recursive estimation problem exists theoretically, but not analytically. This fact is neglected by methods based on local model approximations. In fact, in the global approximation approach, the nonlinear models derived from the underlying physics can be used instead of the linearized models, and the optimal solution, or rather the conditional probability density function $p(x_t|Y_t)$, can be approximated using the Monte Carlo techniques.

One approach among the global approximations is provided by the Sequential Monte Carlo (SMC) methods, also known as the particle filter (Gordon *et al.*, 1993; Kitagawa, 1996; Doucet *et al.*, 2001; Schon, 2006).

This SMC global approach is used in this thesis. In recent years the Sequential Monte Carlo methods have emerged as more effective global approaches and gained more and more ground, both when it comes to the theory and when it comes to the applications. For more references, please refer to Gordon *et al.* (1993), Doucet *et al.* (2001), Doucet *et al.* (2000), Kitagawa (1996), Liu & Chen (1998), Arulampalam *et al.* (2002).

2.2 Kalman Filter Framework

2.2.1 Extended Kalman Filter (EKF)

In the Extended Kalman Filter (EKF) (Smith *et al.*, 1962; Schmidt, 1966), the nonlinear dynamic model and the observational model are linearised around the current estimate, then the standard Kalman Filter is applied. We directly give the EKF algorithm without the detailed proof. For the further and thorough treatment of the EKF, please refer to Jazwinski (1970) and Anderson & Moore (1979).

Algorithm for Extended Kalman Filter

$$x_t^f = \mathscr{M}(x_{t-1}^a) \tag{2.4}$$

$$\mathbf{P}_t^f = \mathbf{M}\mathbf{P}_{t-1}^a\mathbf{M}^T + \mathbf{Q} \tag{2.5}$$

$$\mathbf{K}_t = \mathbf{P}_t^f\mathbf{H}_t^T(\mathbf{H}_t\mathbf{P}_t^f\mathbf{H}_t^T + \mathbf{R}_t)^{-1} \tag{2.6}$$

$$x_t^a = x_t^f + \mathbf{K}_t(y_t^o - \mathscr{H}_t(x_t^f)) \tag{2.7}$$

$$\mathbf{P}_t^a = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t^f \tag{2.8}$$

where $\mathscr{M}$ is the nonlinear dynamic model, $\mathscr{H}$ is the nonlinear measurement model; x_{t-1}^a is the best estimate of the true state at time $t-1$; x_t^f is the forecast of the model state at time t, given only the data available until time $t-1$; $\mathbf{Q}$ is the covariance matrix of the model error; $\mathbf{R}$ is the covariance matrix of the observational error; $\mathbf{P}^f$ is the covariance matrix of the forecast error; $\mathbf{P}^a$ is the covariance matrix of the analysis error; and $\mathbf{K}$ is the Kalman gain matrix. $\mathbf{M}$ and $\mathbf{H}$ are tangent linear models (TLM) of nonlinear models $\mathscr{M}$ and $\mathscr{H}$.

2.2.2 Ensemble Kalman Filter (EnKF)

In the Extended Kalman Filter (EKF), the linearised models ($\mathbf{M}$ and $\mathbf{H}$) are used for the prediction of error statistics.

The Ensemble Kalman Filter (EnKF) is proposed by Evensen (1994) and modified by Burgers *et al.* (1998). In the EnKF, they employ an ensemble of model state members to represent the best estimate of the state variable and error information about its covariance. The ensemble mean states, $\overline{x_i^f}$ and $\overline{x_i^a}$, correspond to the Kalman Filter estimates x^f and x^a. The covariance matrices $\mathbf{P}^f$ and $\mathbf{P}^a$ can be estimated from the spread of the ensembles x_i^f and x_i^a. As the ensemble size becomes larger, the approximation to the Kalman Filter becomes better.

The Algorithm for EnKF (Houtekamer & Mitchell, 2005) as below:

$$x_i^f = \mathscr{M}(x_{i,t-1}^a) + q_i, \qquad i = 1, \ldots, N \tag{2.9}$$

$$q_i \sim N(0, \mathbf{Q}) \tag{2.10}$$

$$\mathbf{P}^f \simeq \mathbf{P}_e^f = \overline{(x^f - \overline{x^f})(x^f - \overline{x^f})^T} \tag{2.11}$$

$$\mathbf{P}^a \simeq \mathbf{P}_e^a = \overline{(x^a - \overline{x^a})(x^a - \overline{x^a})^T} \tag{2.12}$$

$$\mathbf{K} = \mathbf{P}^f \mathscr{H}^T (\mathscr{H} \mathbf{P}^f \mathscr{H}^T + \mathbf{R})^{-1} \tag{2.13}$$

$$y_i^o = y^o + r_i, \qquad i = 1, \ldots, N \tag{2.14}$$

$$r_i \sim N(0, \mathbf{R}) \tag{2.15}$$

$$x_i^a = x_i^f + \mathbf{K}(y_i^o - \mathscr{H} x_i^f), \qquad i = 1, \ldots, N \tag{2.16}$$

The EnKF uses the full nonlinear model $\mathscr{M}$ to transport the error covariances. As can be seen from these equations, given an ensemble of analyses at time $t-1$, the EnKF algorithm yields an ensemble of analyses at time t, that is, EnKF can be performed continuously in time.

2.3 Sequential Monte Carlo (SMC) Methods

Sequential Monte Carlo methods, or particle filter, deal with the problem of recursively estimating the probability density function $p(x_t|Y_t)$. According to the viewpoint of Bayesian statistics, $p(x_t|Y_t)$ contains all the statistical information available about the state variable x_t, based on the information in the measurements Y_t.

The key idea underlying the Sequential Monte Carlo methods is to represent the probability density function $p(x_t|Y_t)$ by a set of samples $\{x_t^{(i)} : i = 1, \dots, M\}$ (also referred to as particles, hence Sequential Monte Carlo methods also known as particle filter) from the probability density function $p(x_t|Y_t)$ and its associated weights. The probability density function $p(x_t|Y_t)$ is approximated with an empirical density function (Schon, 2006),

$$p(x_t|Y_t) \approx \sum_{i=1}^{M} \tilde{q}_t{}^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{M} \tilde{q}_t{}^{(i)} = 1, \qquad \tilde{q}_t{}^{(i)} \geqslant 0, \forall i \tag{2.17}$$

where t denotes time, $\delta(\cdot)$ is the Dirac delta function and $\tilde{q}_t{}^{(i)}$ denotes the weights associated with the particles $x_t^{(i)}$.

The Dirac delta function $\delta(\cdot)$ can be defined as a function on the real line which is zero everywhere except at the origin, where it is infinite,

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \tag{2.18}$$

and which is constrained to satisfy the identity

$$\int_{-\infty}^{+\infty} \delta(x) dx = 1 \tag{2.19}$$

The Dirac delta function $\delta(\cdot)$ has the fundamental property that

$$\int_{-\infty}^{+\infty} f(x)\delta(x-a)dx = f(a) \tag{2.20}$$

2.3.1 Perfect Monte Carlo Sampling

In the perfect Monte Carlo sampling, all the random samples, also known as particles $\{x_t^{(i)} : i = 1, \dots, M\}$ are independent and identically distributed (i.i.d.) from the PDF $p(x_t|Y_t)$, and every sample has equal weight, which is $1/M$. The probability density function can be estimated by these samples according to (2.17). However, it is usually impossible to get i.i.d. samples from the PDF $p(x_t|Y_t)$ at any time t. Nevertheless, the perfect Monte Carlo method shows the key idea in Sequential Monte Carlo (SMC) methods.

2.3.2 Sequential Importance Sampling (SIS)

Unlike the perfect Monte Carlo sampling, all the i.i.d. samples are equally weighted, in Sequential Importance Sampling (SIS), all the i.i.d. samples are weighted according to importance weights $\tilde{q_t}^{(i)}$. In Sequential Importance Sampling, the importance weights $\tilde{q_t}^{(i)}$ contain the information on how probable it is that the corresponding sample was generated from the target PDF $p(x_t|Y_t)$. Sequential Importance Sampling is a more general Monte Carlo method than the perfect Monte Carlo method.

In the SIS implementation, as t increases, the importance weights become more and more skewed and tend to degenerate, which is called the sample impoverishment problem or the weight degeneracy problem. To avoid this weight degeneracy problem, one needs to introduce an additional selection step. In the selection step, the importance weights can be used as the acceptance probabilities, which allows us to generate approximately independent samples $\{\tilde{x}^{(i)}\}_{i=1}^{M}$ from the target density function to be estimated. This implies that the process of generating the samples from the target density function is limited to these samples. More specifically this is realized by resampling among the samples according to

$$Pr(\tilde{x}^{(i)} = x^{(j)}) = \tilde{q}(x^{(j)}), \qquad i = 1, \ldots, M \tag{2.21}$$

where $\tilde{q}(x^{(j)})$ is the weights associated with the particles, and $Pr(\cdot)$ is the probability evaluation.

Resampling step is first introduced in SIS by Rubin (1988) and the modified SIS is renamed after Sampling Importance Resampling (SIR). The SIR algorithm is closely related to the bootstrap procedure, introduced by Efron (1979). This relation is discussed in Smith & Gelfand (1992).

2.3.3 Sequential Monte Carlo Methods/Particle filter

In SMC methods, predicted particles $\{x_{t|t-1}^{(i)}\}_{i=1}^{M}$ are generated from the underlying dynamic model and the filtered particles from the previous time $\{x_{t-1|t-1}^{(i)}\}_{i=1}^{M}$. Conceptually, the predicted particles are obtained simply by passing the filtered particles through the system dynamics. Since the weight function reveals how probable the obtained measurement is given the present state, the more a certain particle explains the received measurement, the more probability that the particle was in fact drawn from the true density. Furthermore, a new set of particles $\{\tilde{x}_{t|t}^{(i)}\}_{i=1}^{M}$ approximating $p(x_t|Y_t)$ is generated by resampling with replacement among the predicted particles$\{x_{t|t-1}^{(i)}\}_{i=1}^{M}$, belonging to the sampling density

$$Pr(\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}(x^{(j)}), \qquad i = 1, \ldots, M \tag{2.22}$$

where $\tilde{q}(x^{(j)})$ is the weights associated with the particles, and $Pr(\cdot)$ is the probability evaluation.

This procedure can be repeated over time, which forms the algorithm of SMC methods. This algorithm was first successfully implemented in practice by Gordon *et al.* (1993). Later it was independently rediscovered by Kitagawa (1996) and Isard & Blake (1998). Further references see Doucet *et al.* (2000), Kitagawa (1996), Liu & Chen (1998), Arulampalam *et al.* (2002).

2.3.3.1 Sequential Monte Carlo Algorithm

This algorithm is used in Gordon *et al.* (1993) and Schon (2006).

Step 1. Initialize the particles, $\{x_{0|-1}^{(i)}\}_{i=1}^{M} \sim p_{x_0}(x_0)$ and set $t := 0$

The particle filter is initialized by drawing samples from the prior density function $p_{x_0}(x_0)$.

Step 2. Measurement update: calculate the importance weights $\{q_t^{(i)}\}_{i=1}^{M}$ according to

$$q_t^{(i)} = p(y_t|x_{t|t-1}^{(i)}), \qquad i = 1, \ldots, M \tag{2.23}$$

and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^{M} q_t^{(j)}$.

In the measurement update, the new measurement is used to assign the probability, represented by the normalized importance weight $\tilde{q}_t^{(i)}$, to each particle. This probability is calculated using the likelihood function $p(y_t|x_{t|t-1}^{(i)})$, which describes how likely it was to obtain the measurement given the information available in the particle.

Step 3. Calculate target probability density function $p(x_t|Y_t)$, according to

$$p(x_t|Y_t) \approx \sum_{i=1}^{M} \tilde{q_t}^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{M} \tilde{q_t}^{(i)} = 1, \qquad \tilde{q_t}^{(i)} \geqslant 0, \forall i \tag{2.24}$$

where t denotes time, $\delta(\cdot)$ is the Dirac delta function and $\tilde{q_t}^{(i)}$ denotes the weights associated with particles $x_t^{(i)}$.

The normalized importance weights and the corresponding particles constitute an approximation of the filtering probability density function $p(x_t|Y_t)$.

Step 4. Resampling: draw M particles, with replacement, according to

$$Pr(\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}(x^{(j)}), \qquad i = 1, \ldots, M \tag{2.25}$$

The resampling step will then return particles which are equally probable.

Step 5. Time update: predict new particles according to

$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)}), \qquad i = 1, \ldots, M \tag{2.26}$$

The time update is just a matter of predicting new particles according to the underlying dynamic model and the filtered particles from the previous time $\{x_{t-1|t-1}^{(i)}\}_{i=1}^{M}$. Conceptually, the predicted particles are obtained simply by passing the filtered particles through the system dynamics.

Step 6. Set $t := t + 1$ and iterate from step 2.

Together with the new observations, these predicted particles form the starting point for another iteration of the assimilation algorithm.

Chapter 3

Assimilation Experiment I: Lorenz 1963 Model

Both Sequential Monte Carlo (SMC) Methods (Gordon *et al.*, 1993) and Ensemble Kalman Filter (EnKF) (Evensen, 1994) are sequential data assimilation methods and of stochastic nature, and both of them rely on Monte Carlo integration of the statistical behavior of the dynamic and measurement model system. Therefore, they have some similarities, and we can make some comparison to investigate the properties of these methods, especially in nonlinear and non-Gaussian dynamics.

It is quite common that the atmospheric and oceanic dynamic systems are nonlinear and non-Gaussian. In this study, we choose the Lorenz model as a test bed (Lorenz, 1963). It describes to some extent the nonlinear and chaotic nature of the atmosphere and ocean.

The renowned Lorenz 1963 model was introduced by Edward Lorenz in 1963, who derived it from the simplified equations of convection rolls arising in the equations of the atmosphere. The Lorenz 1963 model consists of a system of three coupled and nonlinear ordinary differential equations (Lorenz, 1963),

$$\frac{dx}{dt} = \sigma(y - x) \tag{3.1}$$

$$\frac{dy}{dt} = \rho x - y - xz \tag{3.2}$$

$$\frac{dz}{dt} = xy - \beta z \tag{3.3}$$

where, $x(t)$, $y(t)$, and $z(t)$ are the dependent variables, and we have chosen the following commonly used values for the parameters in the equation: $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$.

Our goal is to compare the properties and capabilities of Sequential Monte Carlo (SMC) methods and Ensemble Kalman Filter (EnKF) in strongly nonlinear non-Gaussian dynamics. What is the non-Gaussian dynamics? Let us define the Gaussian dynamics first. In the mathematical theory of probability, a Gaussian process $\{x_t, t \in T\}$ is a stochastic process, of which the probability density function p is normally distributed.

$$p(x_1, x_2, x_3, x_4, \ldots, x_{t-3}, x_{t-2}, x_{t-1}, x_t) \quad \sim \quad N(\mu, \sigma) \tag{3.4}$$

where p is the joint probability density function of the dynamic process, x is the random variable, and t refers to time. If the process is a not Gaussian process, it is a non-Gaussian process.

The Gaussian process is distributed as the normal distribution, also called the Gaussian distribution, which is defined by two parameters, location and scale: the mean and variance (or standard deviation) respectively.

The arithmetic mean is the average, often simply called the mean. The variance is one measure of statistical dispersion, averaging the squared distance of its possible values from the expected value (mean). Whereas the mean is a way to describe the location of a distribution, the variance is a way to capture its scale or degree of being spread out. The unit of variance is the square of the unit of the original variable. The positive square root of the variance, called the standard deviation, has the same units as the original variable.

Mean μ is defined as:

$$\mu = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.5}$$

Standard deviation σ is defined as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (X_i - \mu)^2} \tag{3.6}$$

where μ is the mean of the dynamic process.

For a Gaussian process, the mean and the standard deviation fully characterize the probability density function of the process; however, for non-Gaussian process, the mean and the standard deviation only are insufficient, and higher order moments of the process are needed. Usually the coefficient of skewness and the coefficient of kurtosis are employed.

In probability theory and statistics, the coefficient of skewness is a measure of the asymmetry of the probability distribution. A negative coefficient of skewness means the left tail is longer; the mass of the distribution is concentrated on the right of the figure. The distribution is said to be left-skewed. A positive coefficient of skewness means the right tail is longer; the mass of the distribution is concentrated on the left of the figure. The distribution is said to be right-skewed.

The Coefficient of kurtosis is a measure of the peakedness of the probability distribution. The higher coefficient of kurtosis means more of the variance is due to infrequent extreme deviations, as opposed to the frequent modestly-sized deviations.

The coefficient of skewness γ_1 is defined as:

$$\gamma_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^3}{(\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^2)^{\frac{3}{2}}} \tag{3.7}$$

where μ is the mean of the dynamic process.

The coefficient of kurtosis γ_2 is defined as:

$$\gamma_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^4}{(\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^2)^3} - 3 \tag{3.8}$$

where μ is the mean of the dynamic process.

Through this thesis, we will use the mean μ, the standard deviation σ, the coefficient of skewness γ_1, and the coefficient of kurtosis γ_2 as criteria to compare the assimilation results from EnKF method and SMC methods. Meanwhile, the first quartile, the second quartile (median), the third quartile, and the range of data are employed to check the assimilation results.

We design four different scenarios for the Lorenz 1963 model data assimilation. They are assimilations with observation interval of 0.50, with observation interval of 0.25, with the initial error probability density function of *Beta* Distribution, and with the initial error probability density function of *Gamma* Distribution, see Table 3.1.

Table 3.1: Experiment Design for Lorenz 1963

Assimilation Method	SMC(250 particles) and EnKF(250 ensembles)
Scenario 1	Observation Interval $\delta t_{obs} = 0.50$
Scenario 2	Observation Interval $\delta t_{obs} = 0.25$
Scenario 3	Non-Gaussian Initial Error: Beta
Scenario 4	Non-Gaussian Initial Error: Gamma

3.1 Observation Interval $\delta t_{obs} = 0.50$

The parameters of the Lorenz 1963 model in this case study are $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. In this experiment we choose the same initial conditions as in Miller *et al.* (1994). The initial condition (x, y, z) is given by (1.508870, -1.531271, 25.46091), and the integration duration of the experiment is 50 dimensionless time units, with an integration time step of 0.01. The true value (reference resolution) is created by integrating the model with the above configurations.

The distance between two nearest measurements is $\delta t_{obs} = 0.50$ and observations are made on the x, y and z coordinates. In this case study, the system initial error is Gaussian $N(0.0, 2.0)$, and the observational error is also Gaussian $N(0.0, 2.0)$. The observations are simulated by adding normally distributed noise with zero mean and variance equal to 2.0 to the true value (reference solution). Initial conditions are also simulated by adding normally distributed noise with zero mean and variance equal to 2.0 to the true value (reference solution). This system of equations is integrated by Numerical Algorithms Group (NAG) Numerical Libraries with the fourth-order Runge-Kutta method. The assimilation experiments are run on an SGI Altix 3000 (64 Intel Itanium - 2 1500 MHz CPUs) global shared memory supercomputer.

The filter performance will be evaluated by three factors: 1) root mean square error ($RMSE$); 2) CPU computation time; 3) statistics of the probability density function (PDF), which is estimated from the true resolution and assimilated estimates.

The root mean square error ($RMSE$) is calculated between the reference solution and the filtering estimate (analysis) averaged over the whole assimilation period.

We performed both the SMC methods and the EnKF method data assimilation in the Lorenz 1963 model, with different numbers of SMC particles and EnKF ensemble members. The number of SMC particles is 250. The number of EnKF ensemble members is also 250.

The assimilation for x, y, and z is performed. However, the assimilation method is independent of state variables, thus the assimilation results for three variables x, y, and z are quite similar. Therefore, only the assimilation result for x is showed in this thesis.

Table 3.2: Computation time and RMSE for Lorenz 1963 (Case: $\delta t_{obs} = 0.50$)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	4.830	11.246
RMSE (X)	1.8520	2.0208
RMSE (Y)	2.9850	3.2572
RMSE (Z)	2.7383	2.9262

Table 3.3: Statistics of PDF of x-component of Lorenz 1963 (Case: $\delta t_{obs} = 0.50$)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	0.6415	0.7211	0.8688	-0.0795	-0.2272
Standard deviation	7.8752	7.9397	8.0119	-0.0645	-0.1366
Coefficient of skewness	-0.1544	-0.1766	-0.2073	0.0222	0.0528
Coefficient of kurtosis	-0.6077	-0.6330	-0.7009	0.0252	0.0931
First Quartile	-4.3857	-4.3494	-4.4666	-0.0363	0.0809
Second Quartile	1.0567	0.8517	1.5545	0.2050	-0.4978
Third Quartile	5.9689	6.2736	6.5900	-0.3047	-0.6212
Range	35.4200	35.8230	36.0129	-0.4030	-0.5929

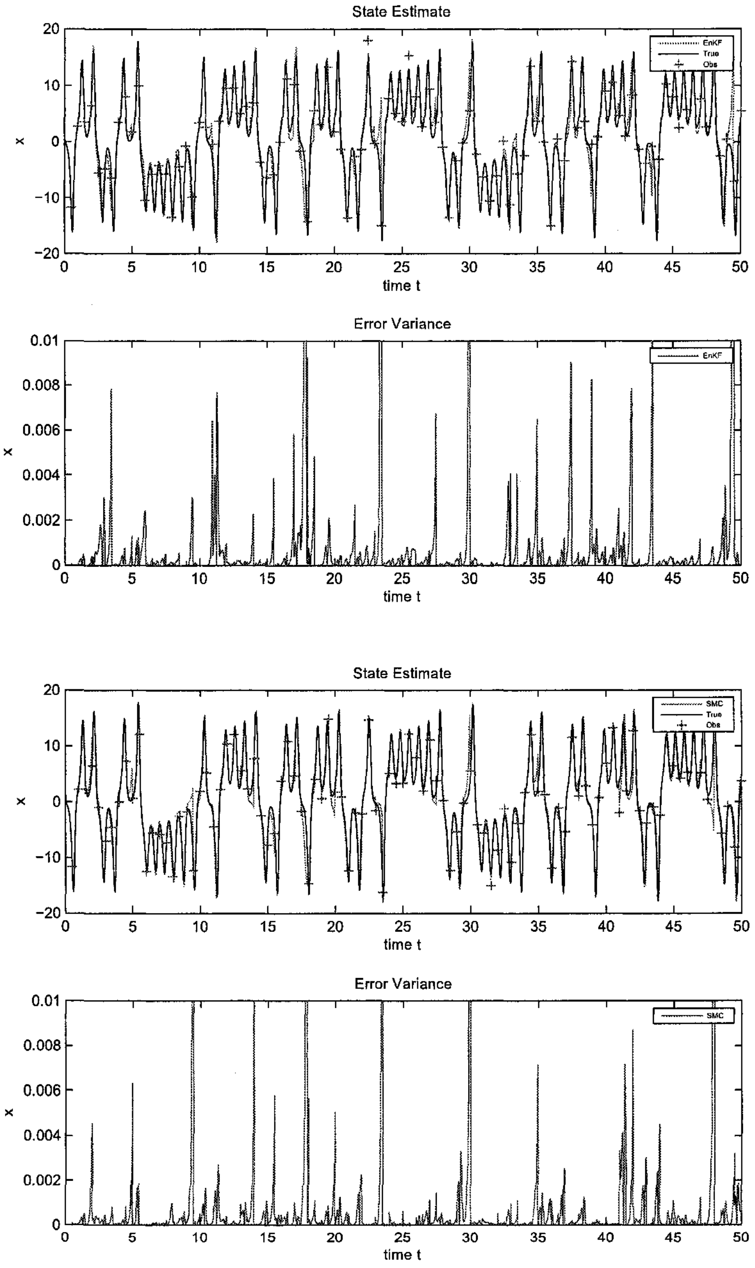


Figure 3.1: State estimate and error variance of x-component of Lorenz 1963 model for EnKF and SMC methods with filter size of 250, $\delta t_{obs} = 0.50$

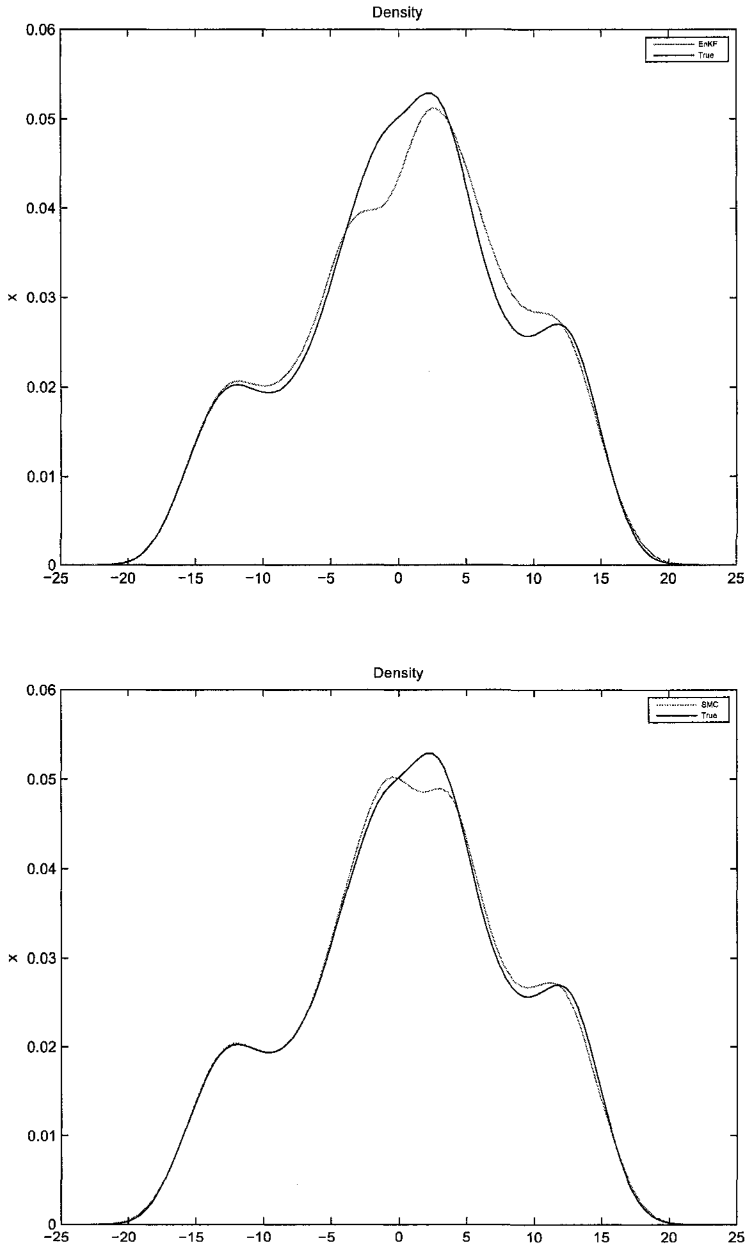


Figure 3.2: Probability density function of x-component of Lorenz 1963 model for EnKF and SMC methods with filter size of 250, $\delta t_{obs} = 0.50$

Figure 3.1 shows the variable x estimate and the error variance estimate with time step from both SMC methods and EnKF method with 250 particles and ensemble members. The error variance which defined as in Evensen (1997) is scaled by N where N is a scalar quantity of time steps of the total assimilation period

$$Error\ Variance = \frac{1}{N}(X_t^{estimate} - X_t^{true})^2 \tag{3.9}$$

Both the SMC methods and the EnKF method do reasonably good jobs in tracking the phase transitions and also in reproducing the correct amplitudes of the reference solution. There are some locations where the filter estimates start to diverge from the reference solution. To compare the assimilation results, we divide the total assimilation period into two, the first half and the second half, so that we want to examine whether the assimilation results become better with more observations coming into the data assimilation system. Meanwhile, since the error variance is already rescaled, the error variance in all figures is just used to compare its relatively range within two methods, EnKF and SMC methods. To compare the $RMSE$ variation with time, we choose 0.006 as a standard. For the EnKF method, at time 3, 11, 17, 23, 27, 30, 35, 37, 39, 42, 44, and 49, the error variance is greater than 0.006, which means the filter estimates deviate from the true solution. Among these locations, 8 out of 12 are in the second half of the assimilation period. For SMC methods, at time 5, 9, 14, 17, 23, 30, 35, 42, 43, and 47, the error variance is greater than 0.006. Among these locations, 5 out of 10 are in the second half of the assimilation period. Despite these divergences, both methods recover quickly and track the reference solution again.

Table 3.2 indicates the CPU computation time and the $RMSE$ for both methods in this case. The CPU computation time is 4.830 s for SMC methods, while 11.246 s for EnKF method. The EnKF method takes almost twice longer than the SMC methods. The $RMSE$ is 1.8520, 2.9850, and 2.7383 for x, y, and z for SMC methods, while it is 2.0208, 3.2572, and 2.9262 for x, y, and z for EnKF method. SMC methods are slightly better than the EnKF method in this case.

Figure 3.2 shows the probability density function of x component of the Lorenz 1963 dynamic system for both EnKF and SMC methods. The probability density function is calculated by the kernel density estimation method (Parzen, 1962 and Silverman, 1986). In Matlab, the kernel density estimation is implemented through the *ksdensity* function. In this thesis, all probability density functions are estimated by Matlab *ksdensity* function. From Figure 3.2, we can see clearly that the probability density functions are non-Gaussian. Both methods can assimilate it quite well, but, they both have some difficulties to reach the exact peaks of the probability density function.

Table 3.3 shows the statistics of the probability density function of x component of the Lorenz 1963 model. The mean, the standard deviation, the coefficient of skewness, and the coefficient of kurtosis for the SMC methods are closer to those of the true state than those of the EnKF method. The quartiles and range from the SMC methods are closer to those of the true resolution than those of the EnKF method. Therefore, the SMC methods estimate the probability density function slightly better than the EnKF in this case.

3.2 Observation Interval $\delta t_{obs} = 0.25$

In this second experiment, the experimental setup is the same as the previous one, except that the observation interval between two measurements decreases from $\delta t_{obs} = 0.50$ to $\delta t_{obs} = 0.25$ for this case. That means we have more observations in the assimilation process in case 2 than in case 1.

Table 3.4: Computation time and RMSE for Lorenz 1963 (Case: $\delta t_{obs} = 0.25$)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	5.309	18.169
RMSE (X)	1.4003	1.0156
RMSE (Y)	2.1914	1.6157
RMSE (Z)	1.9663	1.5755

Table 3.5: Statistics of PDF of x-component of Lorenz 1963 (Case: $\delta t_{obs} = 0.25$)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	0.6415	0.7080	0.7562	-0.0664	-0.1146
Standard deviation	7.8752	7.9889	7.8370	-0.1137	0.0382
Coefficient of skewness	-0.1544	-0.1200	-0.1785	-0.0340	0.0240
Coefficient of kurtosis	-0.6077	-0.6853	-0.6307	0.0775	0.0229
First Quartile	-4.3857	-4.8548	-4.3910	0.4691	0.0054
Second Quartile	1.0567	1.3272	1.2496	-0.2705	-0.1929
Third Quartile	5.9689	6.1060	6.0626	-0.1371	-0.0937
Range	35.4200	35.3180	34.4886	0.1020	0.9314

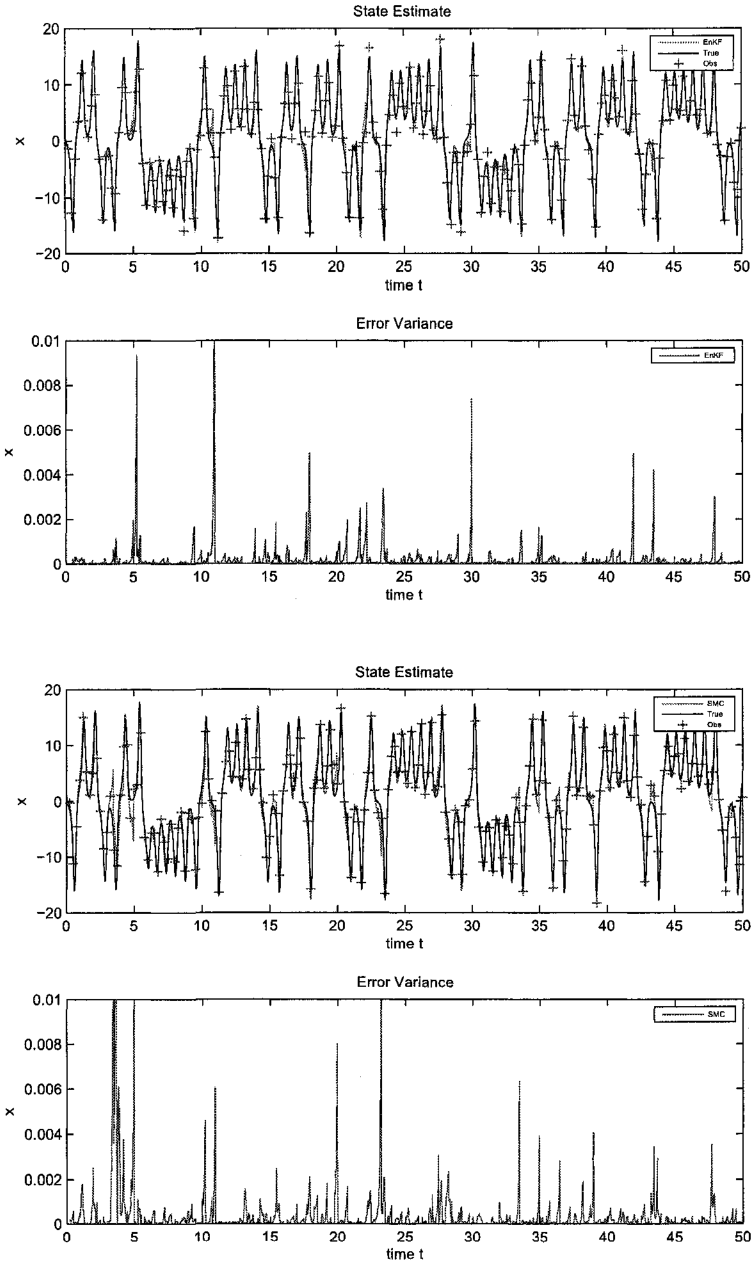


Figure 3.3: State estimate and error variance of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, $\delta t_{obs} = 0.25$

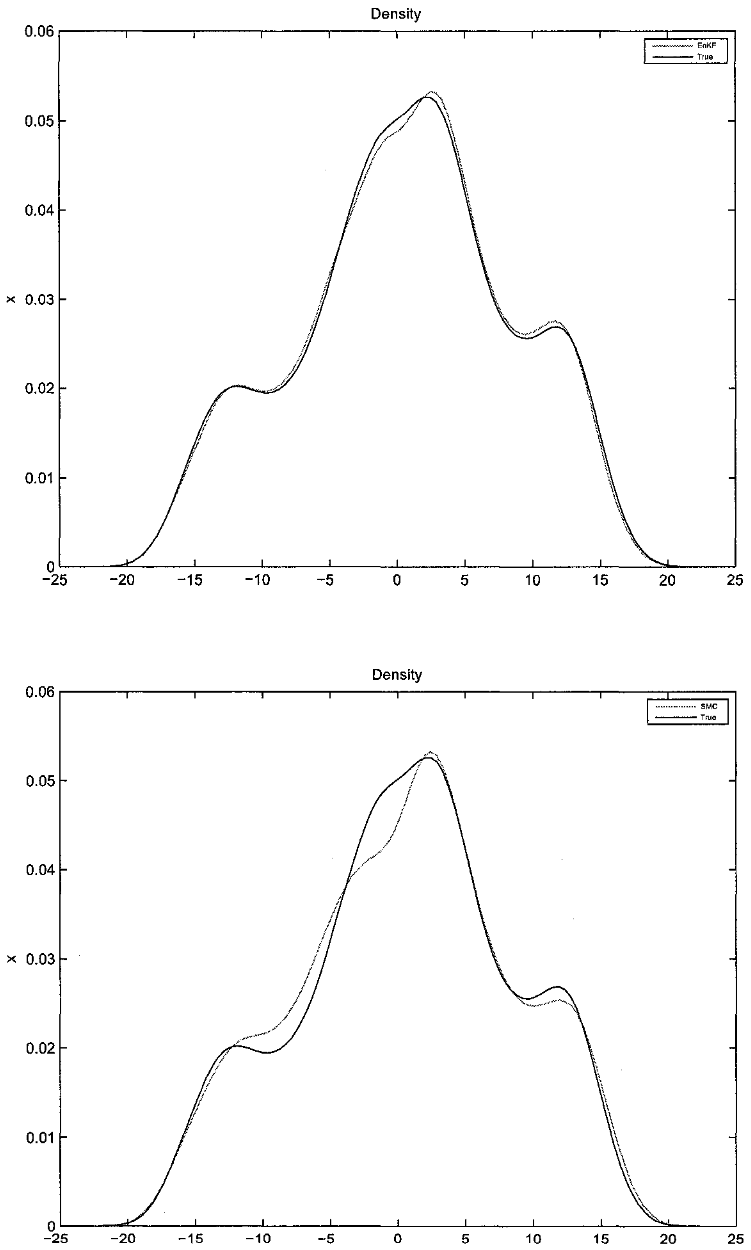


Figure 3.4: Probability density function of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, $\delta t_{obs} = 0.25$

Figure 3.3 shows the variable x estimate and the error variance estimate with time step from both the SMC methods and the EnKF method with the filter size of 250. In tracking the phase transitions, there are some locations where the filter estimates diverge from the reference solution. For EnKF method, at time 5, 11, and 30, the error variance is greater than 0.006, which means filter estimates deviate from true solution. Among these locations, 1 out of 3 is in the second half of the assimilation period. For SMC methods, at time 4, 5, 11, 20, 24, and 34, the error variance is greater than 0.006. Among these locations, 1 out of 6 is in the second half of the assimilation period. With more observations, the EnKF method outperforms the SMC methods. For both methods, the transition in the second half becomes smoother than that in the first half. Despite these divergences, both methods recover quickly and track the reference solution again. From the error variance with time, we can see the error variance decreases with time in this case.

Table 3.4 indicates the CPU computation time and the $RMSE$ for both methods in the case. The CPU computation time is 5.309 s for SMC methods, while it is 18.169 s for EnKF method. The EnKF method takes almost 3 times longer than the SMC methods. The $RMSE$ is 1.4003, 2.1914, and 1.9663 for x, y, and z for SMC methods, while it is 1.0156, 1.6157, and 1.5755 for x, y, and z for EnKF method. The EnKF method is significantly better than the SMC methods in this case with more observations available.

Figure 3.4 shows the probability density function of x component of the Lorenz 1963 dynamic system for both the EnKF method and the SMC methods. From Figure 3.4, we can see clearly that the probability density function is non-Gaussian. Both methods can assimilate this nonlinear dynamic process quite well; however, the EnKF method almost reaches the exact peak of the probability density function, which is better than the SMC methods.

Table 3.5 shows the statistics of the probability density function of x component of Lorenz 1963 model. The mean for the SMC methods are closer to the true resolution than that of the EnKF method, while the standard deviation, the coefficient of skewness, and the coefficient of kurtosis for the EnKF method are closer to the true state than the SMC methods, as well as the quartiles and range of the data.

In Table 3.2 and Table 3.4, EnKF takes more than twice the time than the SMC methods do. That means both methods can achieve reasonably good results, but the SMC methods is more efficient than EnKF in this case. The EnKF algorithm used in this thesis is explained in Evensen (2003). We need to perform an analysis algorithm to each individual member, which is why EnKF takes much more time than SMC methods. In addition to the reason above, the EnKF analysis algorithm requires the calculation of the inverse of matrix, which is quite time consuming. One way to reduce the computational time for the EnKF is to reduce the ensemble size. For this, one possible option is to replace random perturbation in Kalman Filter by a deterministic perturbation, which turns out to be Unscented (Sigma-Point) Kalman Filter (Julier & Uhlmann, 1996).

3.3 Non-Gaussian Initial Error: Beta

For the third case, we keep the experimental setup the same as the first one; expect that the system initial error is non-Gaussian, Beta Distribution. In Case Beta, Beta (2.0, 5.0) is used as the initial probability density function for model integration. In the first two cases, the model starts with a Gaussian error, and the probability density function may become non-Gaussian after the model iteration starts. In this case, in the beginning, the model starts with a non-Gaussian probability density function, and it will remain non-Gaussian after the model iteration.

The EnKF method always uses Gaussian marginal probability density function to represent non-Gaussian marginal probability density function during the assimilation process; theoretically it is not sufficient, because only lower-order moments (mean, variance) are considered. While SMC methods directly estimate non-Gaussian marginal density function, theoretically it is much better than EnKF.

Figure 3.5 shows the variable x estimate and the error variance estimate with time from both SMC methods and EnKF method with 250 particles and ensemble members. In tracking the phase transitions there are some locations where the filter estimates diverge from the reference solution. In spite of these divergences, both methods recover quickly and track the reference solution again. For the

EnKF method, at time 2, 5, 6, 11, 16, 23, 30, 37, 43, 44 and 45, the error variance is greater than 0.006, which shows filter estimates deviate from true solution. Among these locations, 5 out of 11 are in the second half of the assimilation period. For the SMC methods, at time 2, 3, 6, 9, 11, 16, 17, 23, 27, 30, 37, and 44, the error variance is greater than 0.006. Among these locations, 4 out of 12 are in the second half of the assimilation period, which indicates that the second half assimilation transition for the SMC method is smoother than that for the EnKF method.

Table 3.6 indicates the CPU computation time and the $RMSE$ for both methods in the case. The CPU computation time is 4.877s for SMC methods, while it is 11.297s for EnKF method. The $RMSE$ is 2.1640, 3.3007, and 3.8650 for x, y, and z for SMC methods, while it is 2.3394, 3.5670, and 2.9842 for x, y, and z for EnKF method. For variables x and y, the SMC methods is better, while for variables z, the EnKF method is better.

Figure 3.6 shows the probability density function of the x component of the Lorenz 1963 dynamic system for both EnKF and SMC methods. From Figure 3.6, we can see clearly that the probability density function is non-Gaussian, which has one major peak and two weak peaks. Both methods can assimilate it reasonably well, but both of them have trouble to track the exact peaks of the probability density function.

Table 3.7 shows the statistics of the probability density function of x of Lorenz 1963 model. The EnKF assimilated result is closer to the true resolution than SMC methods in the mean, the coefficient of skewness, the median, the third quartile and the range; while the SMC methods are better in the standard deviation, the coefficient of kurtosis, and the first quartile than the EnKF method.

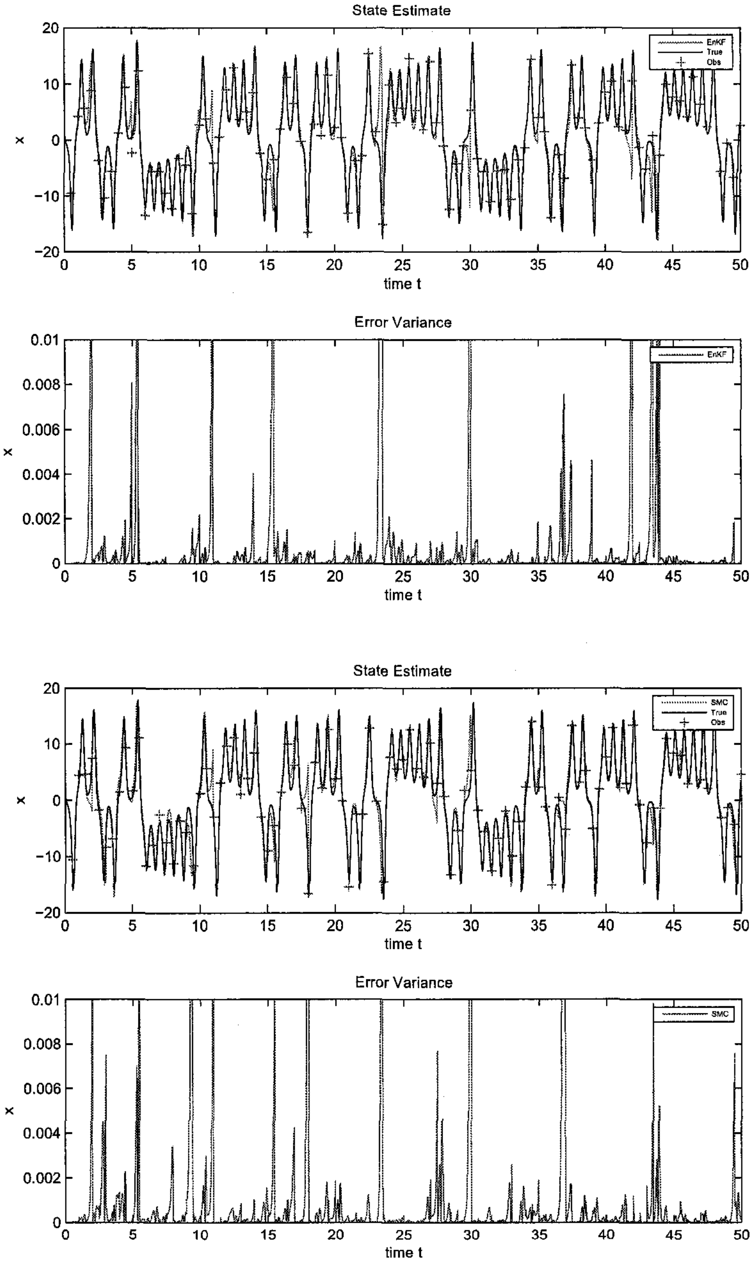


Figure 3.5: State estimate and error variance of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, Initial Beta distribution

Table 3.6: Computation time and RMSE for Lorenz 1963 (Case: Beta)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	4.877	11.297
RMSE (X)	2.1640	2.3394
RMSE (Y)	3.3007	3.5670
RMSE (Z)	3.8650	2.9842

Table 3.7: Statistics of PDF of x-component of Lorenz 1963 (Case: Beta)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	0.6415	0.9465	0.7676	-0.3049	-0.1260
Standard deviation	7.8752	7.9873	8.0014	-0.1120	-0.1262
Coefficient of skewness	-0.1544	-0.2291	-0.1659	0.0747	0.0115
Coefficient of kurtosis	-0.6077	-0.6902	-0.7426	0.0824	0.1348
First Quartile	-4.3857	-4.4127	-4.7078	0.0270	0.3221
Second Quartile	1.0567	1.6845	1.3963	-0.6278	-0.3396
Third Quartile	5.9689	6.7403	6.5498	-0.7714	-0.5809
Range	35.4200	34.7577	35.2279	0.6623	0.1921

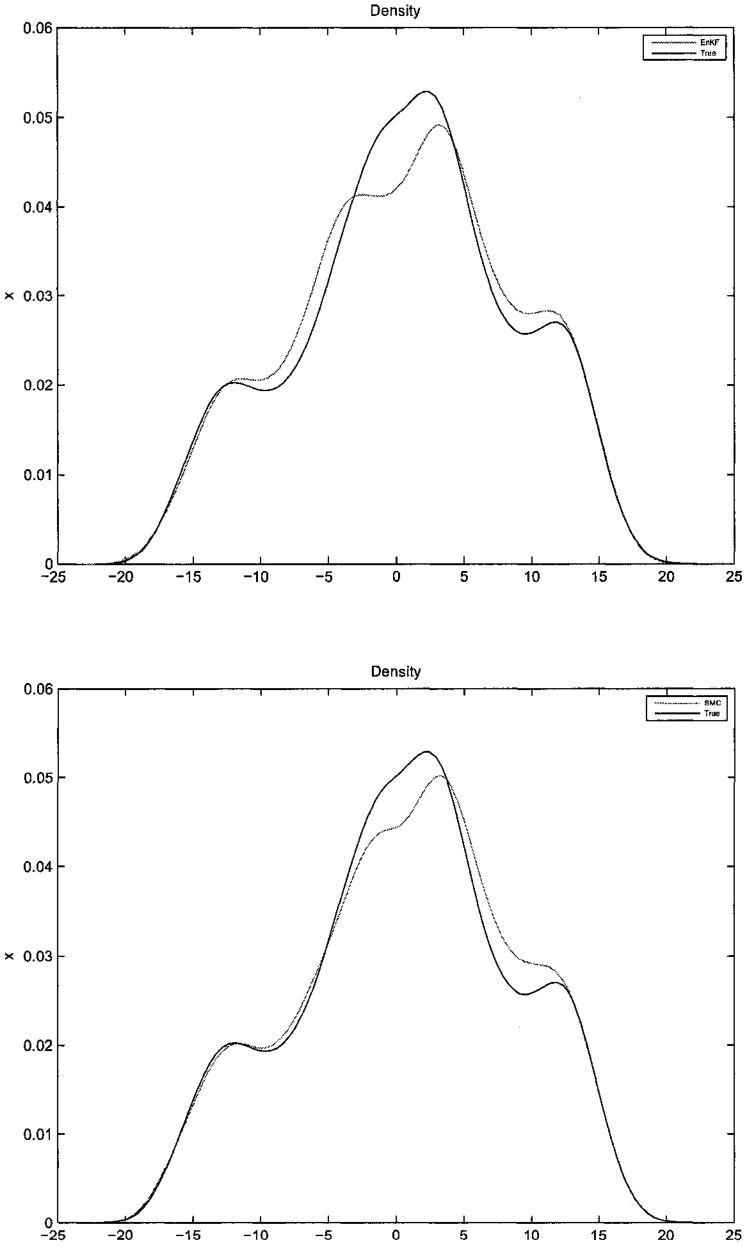


Figure 3.6: Probability density function of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, Initial Beta distribution

3.4 Non-Gaussian Initial Error: Gamma

For the fourth case, we keep the experimental setup the same as the third one; expect that the system initial error is non-Gaussian, Gamma Distribution. In Case Gamma, Gamma (2.0, 2.0) is used to integrate the model forward as the initial error probability density function.

Figure 3.7 shows the variable x state estimate and the error variance estimate variation with time forward for both the SMC methods with 250 particles and the EnKF method with 250 ensemble members. In tracking the phase transitions there are some locations where the filter estimates diverge from the reference solution. For example, for the EnKF method, at time 4, 5, 18, 24, 30, 36, 37, 44, and 49, the error variance is greater than 0.006, which indicates filter estimates deviate from true solution. Among these locations, 5 out of 9 are in the second half of the assimilation period, or rather, the estimate deviation occurs all through the assimilation process. For the SMC methods, at time 2, 4, 5, 9, 17, 20, 21, 24, 37, 39, 44, and 49, the error variance is greater than 0.006. Among these locations, 4 out of 12 are in the second half of the assimilation period, which shows the second half assimilation is better than the first half. Despite these divergences, both methods recover quickly and track the reference solution again. In general, both the SMC methods and the EnKF method do the case experiment well.

Table 3.8 indicates the CPU computation time and the $RMSE$ for both methods in the case. The CPU computation time is 4.925 s for SMC methods, while it is 11.295 s for EnKF method. The EnKF method takes almost as three times long as the SMC methods. The $RMSE$ is 2.5674, 4.0898, and 3.7482 for x, y, and z for SMC methods, while it is 2.3808, 3.5195, and 3.4267 for x, y, and z for EnKF method. The EnKF method is slightly better than that of the SMC methods in this case with Gamma Initial Distribution.

Figure 3.8 shows the probability density function of the x component of the Lorenz 1963 dynamic system for both EnKF and SMC methods. From Figure 3.8, we can see clearly that the probability density function is non-Gaussian, since it has one major peak and another two small peaks. Both methods can assimilate it reasonably well, but neither of them can reach the exact peaks of the probability density function. Based on the 4 cases above, the process probability density functions are quite similar to one another in the 4 experiments, which indicate that the process PDF is mainly determined by dynamics itself. Still we could try to find the optimal initial probability density function for the specific data assimilation sytem.

Table 3.9 shows the statistics of the probability density function of x of the Lorenz 1963 model. Both the mean for the SMC methods and the EnKF method are slightly larger than the true resolution; the standard deviation for the SMC methods is smaller than that of the true resolution, while the standard deviation for the EnKF method is slightly larger than that of the true resolution. the coefficients of skewness are -0.1544, -0.1819, and -0.1897; the difference of coefficients of kurtosis between true resolution and assimilated estimate for the EnKF method is 10 times larger than that of the SMC methods, both absolute values are quite small though. Moreover, all the quartiles in the EnKF method are better than that of the SMC methods, except the range in the SMC methods are better.

The SMC methods have the theoretical advantage, why do the experimental results show similar estimates? Why do not the SMC methods outperform EnKF methods?

$p(x_1, x_2, \ldots, x_n)$ is the probability density function of the dynamic process, which is non-Gaussian. The marginal probability density function $p(x_n)$ is estimated through the data assimilation process is also non-Gaussian. In EnKF, we assume all the marginal probability density function is Gaussian, which is not true for non-Gaussian dynamics, no matter whether it is linear or nonlinear dynamics, and the mean and covariance of marginal probability density is used to fully characterize the dynamics. However, in SMC methods, the marginal probability density function is estimated directly from sample particles.

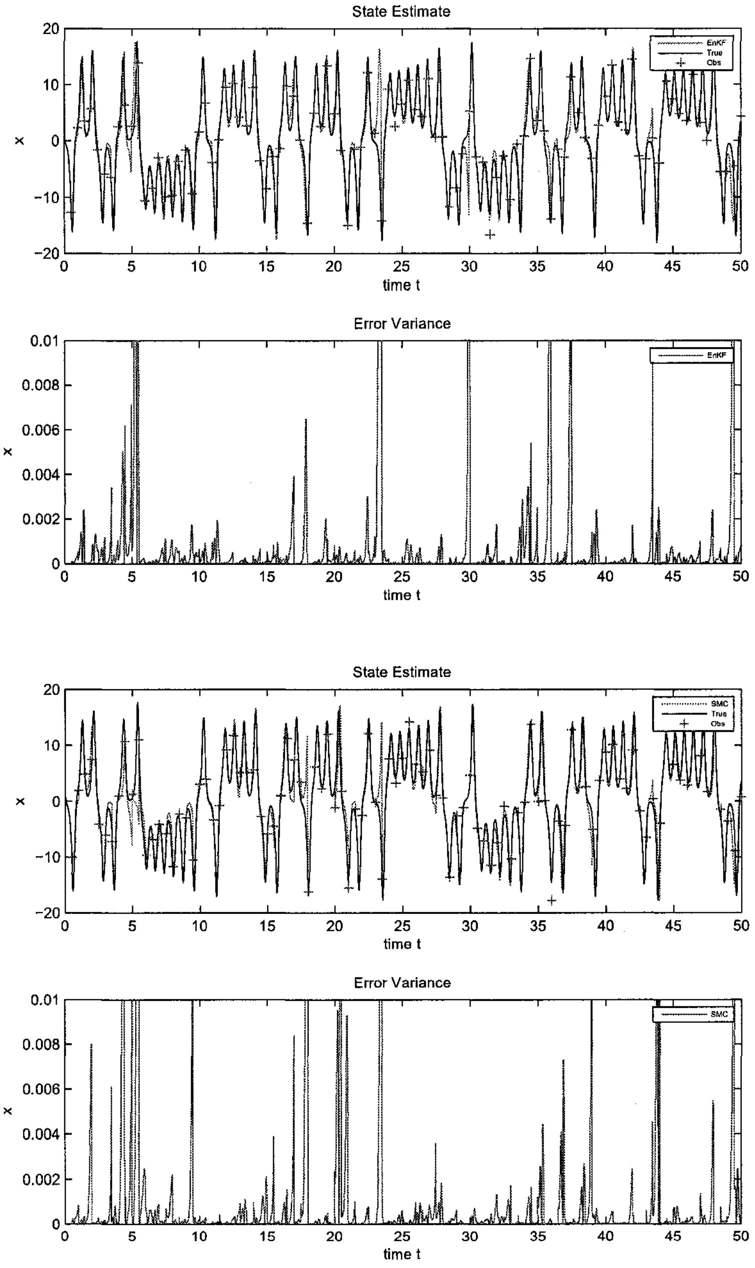
The $RMSE$ is calculated from true resolution and assimilation resolution, it is only mean value or rather the first order of the moment of marginal probability density function. EnKF employs Gaussian marginal probability density function to represent non-Gaussian marginal probability density functions, which is not sufficient theoretically. However, it may be sufficient to represent the mean value of marginal probability density function. That is why $RMSE$ for both EnKF and SMC are quite similar. Since we do not have a true non-Gaussian marginal probability density function as a reference, it is difficult to verify whether the marginal density function in SMC methods can fully represent true dynamic marginal probability density function or not.

Table 3.8: Computation time and RMSE for Lorenz 1963 (Case: Gamma)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	4.925	11.295
RMSE (X)	2.5674	2.3808
RMSE (Y)	4.0898	3.5195
RMSE (Z)	3.7482	3.4267

Table 3.9: Statistics of PDF of x-component of Lorenz 1963 (Case: Gamma)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	0.6415	0.7903	0.7485	-0.1487	-0.1069
Standard deviation	7.8752	7.7757	7.9799	0.0995	-0.1047
Coefficient of skewness	-0.1544	-0.1819	-0.1897	0.0275	0.0353
Coefficient of kurtosis	-0.6077	-0.5980	-0.6958	-0.0097	0.0880
First Quartile	-4.3857	-4.2855	-4.5641	0.1785	-0.1002
Second Quartile	1.0567	1.0532	1.2945	-0.2378	0.0035
Third Quartile	5.9689	6.2274	6.4560	-0.4871	-0.2585
Range	35.4200	35.6834	35.0820	-0.2634	0.3380



State Estimate

Error Variance

State Estimate

Error Variance

Figure 3.7: State estimate and error variance of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, Initial Gamma distribution

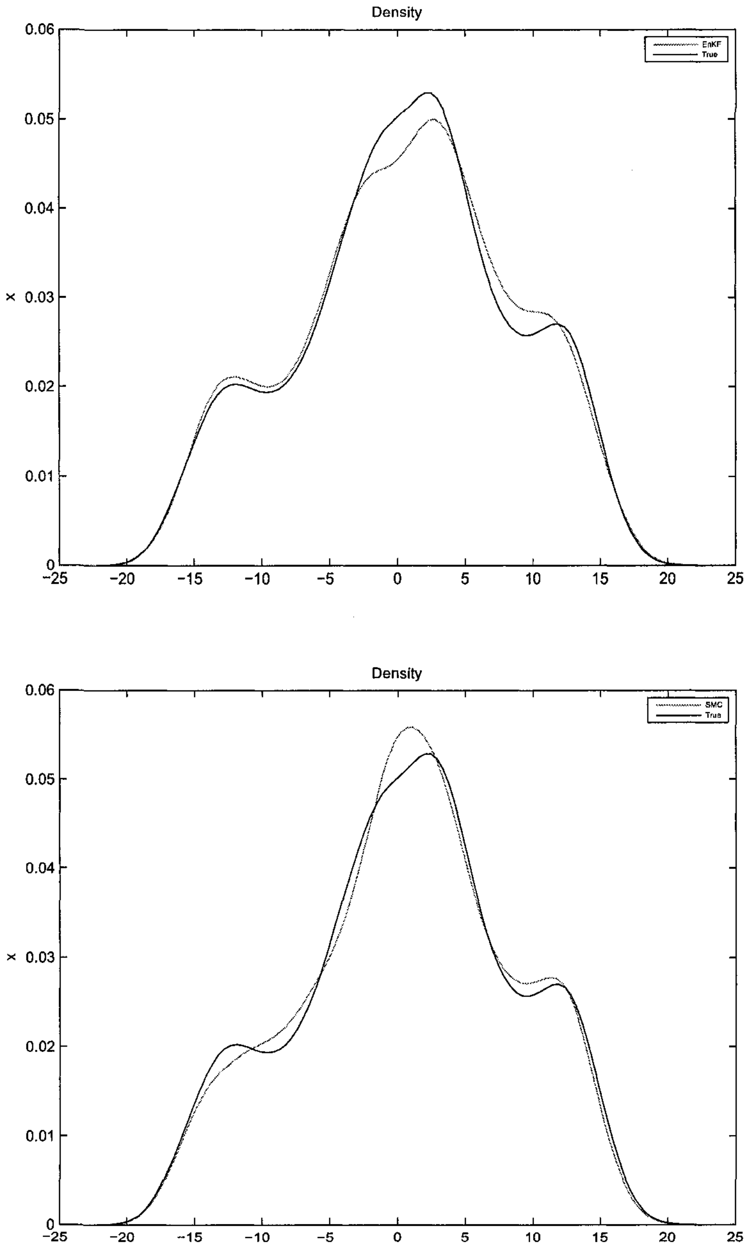


Figure 3.8: Probability density function of x-component of Lorenz 1963 model for EnKF and SMC methods, Filter size = 250, Initial Gamma distribution

Chapter 4

Assimilation Experiment II: Lorenz 1996 Model

The second experimental design employs the Lorenz 1996 model as the test bed. The Lorenz 1996 model (Lorenz, 1996) represents an atmospheric variable X at J equally spaced points around a circle of the constant latitude. The jth component is propagated forward in time following the differential equation

$$\frac{dX_j}{dt} = (X_{j+1} - X_{j-2})X_{j-1} - X_j + F \tag{4.1}$$

where $j = 0, \ldots, J-1$ represents the spatial coordinates (longitude). F is a constant external forcing term, which indicates the dynamics is weakly chaotic when $F = 5$ or $F = 6$, it is highly chaotic when $F = 8$, and it is fully turbulent when $F = 16$. Note that this model is not a simplification of any atmospheric system, however, it is designed to satisfy three basic properties: it has linear dissipation (the $-X_j$ term) that decreases the totally energy , an external forcing term F that can increase or decrease the total energy and a quadratic advection term that conserves the total energy just like many atmospheric models. In its configuration, $J = 40$ variables and boundary conditions are cyclic, i.e. $X_{-1} = X_{j-1}$, $X_0 = X_j$, and $X_{j+1} = X_1$, which means the distance between two adjacent grid points roughly represents the midlatitude Rossby radius (about 800 km), assuming the circumference of the midlatitude belt is about 30000 km (Majda & Harlim, 2008).

In this experimental design, we define three different forcing term F scenarios. The first category is $F = 5$, which indicates that the model is weakly chaotic, the second category is $F = 8$, which indicates that the model is highly chaotic, and the last category is $F = 16$, which indicates that the model is fully turbulent, see Table 4.1.

This dynamic model is integrated by the Numerical Algorithms Group (NAG) Numerical Libraries with the fourth-order Runge-Kutta method, and the integration time step of 0.05, corresponding to 6 hours in the realistic atmospheric physics. The initial condition is given after a spin up integration for 10 years. The duration of the experiment setup is 40 dimensionless time units. The observation interval between two measurements is $\delta t_{obs} = 0.50$ and observations are available for all 40 variables. In this case study, the system initial error is Gaussian $N(0.0, 2.0)$, and observational error is also Gaussian $N(0.0, 2.0)$. The observations are simulated by adding normally distributed noise with zero mean and variance equal to 2.0 to the reference solution. Initial conditions are simulated by adding normally distributed noise with zero mean and variance equal to 2.0 to reference solution.

The filter performance will also be evaluated by the root mean square error ($RMSE$) between the true value (reference solution) and the filtering estimate averaged over the whole assimilation period, the CPU computational time, and the statistics of the probability density functions. The assimilation experiments run on an SGI Altix 3000 (64 Intel Itanium - 2 1500 MHz CPUs) global shared memory supercomputer.

We performed both the SMC methods and the EnKF method data assimilation in the Lorenz 1996 model, with different numbers of SMC particles and EnKF ensemble members equal 250.

For the Lorenz 1996 model, 40 variables are functionally equal. Also the assimilation method is independent of model state variables. Therefore, only the assimilation result for state variable $X1$, $X20$, and $X30$ are shown in this thesis.

Table 4.1: Experiment Design for Lorenz 1996

Assimilation Method	SMC(250 particles) and EnKF(250 ensembles)
Scenario 1	Weakly Chaotic $F = 5$
Scenario 2	Highly Chaotic $F = 8$
Scenario 3	Fully Turbulent $F = 16$

4.1 Weakly Chaotic $F = 5$

In Fig 4.1, both the SMC methods and the EnKF method perform reasonably well in tracking the phase transitions and also in reproducing the correct amplitudes of the reference solution. We divide the whole assimilation period into two. Both the SMC methods and the EnKF method take almost half of the whole period to start to track the reference solution accurately. When the dynamics are weakly chaotic, both methods can assimilate the dynamic process well and quickly, since the second half is obviously better than the first half. However, there are some locations where the filter estimates start to diverge from the reference solution. For the EnKF method, at time 3, 12, and 13, the error variance is greater than 0.006, which means filter estimates deviate from true solution. Among these locations, none of 3 is in the second half of the assimilation period. For the SMC methods, at time 3, 4, 16, 22, and 33, the error variance is greater than 0.006. Among these locations, 2 out of 10 are in the second half of the assimilation period. Despite these divergences, both methods recover quickly and track the reference solution again. From the error variance variation with time, we can see the strong error growth at those phase transition locations. Furthermore, the noisy level of $RMSE$ for the EnKF method is lower than that for the SMC methods.

Table 4.2 indicates the CPU computation time and the $RMSE$ for both methods in the case. The CPU computation time is 15.809 s for the SMC methods, while it is 142.960 s for the EnKF method. The EnKF method takes almost 9 times longer than the SMC methods. The $RMSE$ is 1.2777, 1.1791, and 1.1207 for $X1$, $X20$, and $X30$ for the SMC methods, while it is 0.9404, 0.9297, and

0.9310 for $X1$, $X20$, and $X30$ for the EnKF method. The SMC methods are slightly worse than the EnKF method in this case.

From Figure 4.2, we can see clearly that the probability density function of $X1$ is non-Gaussian, not strongly non-Gaussian though. Both methods can assimilate it quite well, but, they both have some difficulties to reach the exact peaks of the probability density function.

Table 4.3 shows the statistics of the probability density function of $X1$ of the Lorenz 1996 model. The mean and the standard deviation for the SMC methods assimilated result are closer to the true state than that of the EnKF method. However, for the higher order moments of the probability density function, the coefficient of skewness for the SMC methods is closer to the true resolution than the EnKF method, while the coefficient of kurtosis for the EnKF method is closer to the true resolution than the SMC methods. All three quartiles in the SMC methods are better than those in the EnKF method except the range.

Table 4.2: Computation Time and RMSE of Lorenz 1996 ($F = 5$)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	15.809	142.960
RMSE (X1)	1.2777	0.9404
RMSE (X20)	1.1791	0.9297
RMSE (X30)	1.1207	0.9310

Table 4.3: Statistics of PDF of $X1$ of Lorenz 1996 ($F = 5$)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	1.6262	1.6123	1.6853	0.0138	-0.0591
Standard deviation	2.3589	2.4508	2.2390	-0.0918	0.1198
Coefficient of skewness	0.0602	0.0438	-0.0397	0.0164	0.1000
Coefficient of kurtosis	-0.6937	-0.4271	-0.6882	-0.2666	-0.0055
First Quartile	-0.0816	-0.0890	0.0664	0.0074	-0.1480
Second Quartile	1.5014	1.5316	1.6122	-0.0302	-0.1108
Third Quartile	3.3845	3.4336	3.5181	-0.0491	-0.1336
Range	11.0311	13.9137	11.5739	-2.8826	-0.5428

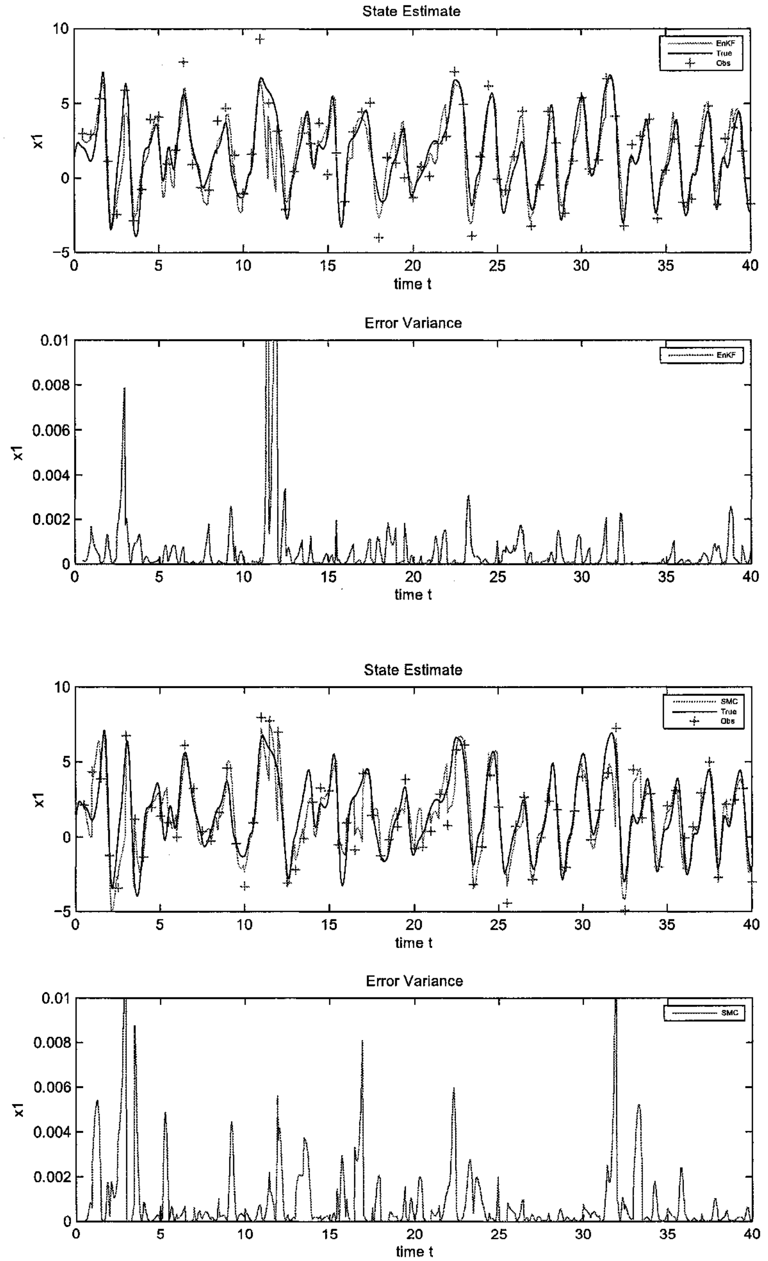


Figure 4.1: State estimate and error variance of $X1$ of Lorenz 1996 model ($F = 5$) for EnKF and SMC methods with filter size of 250

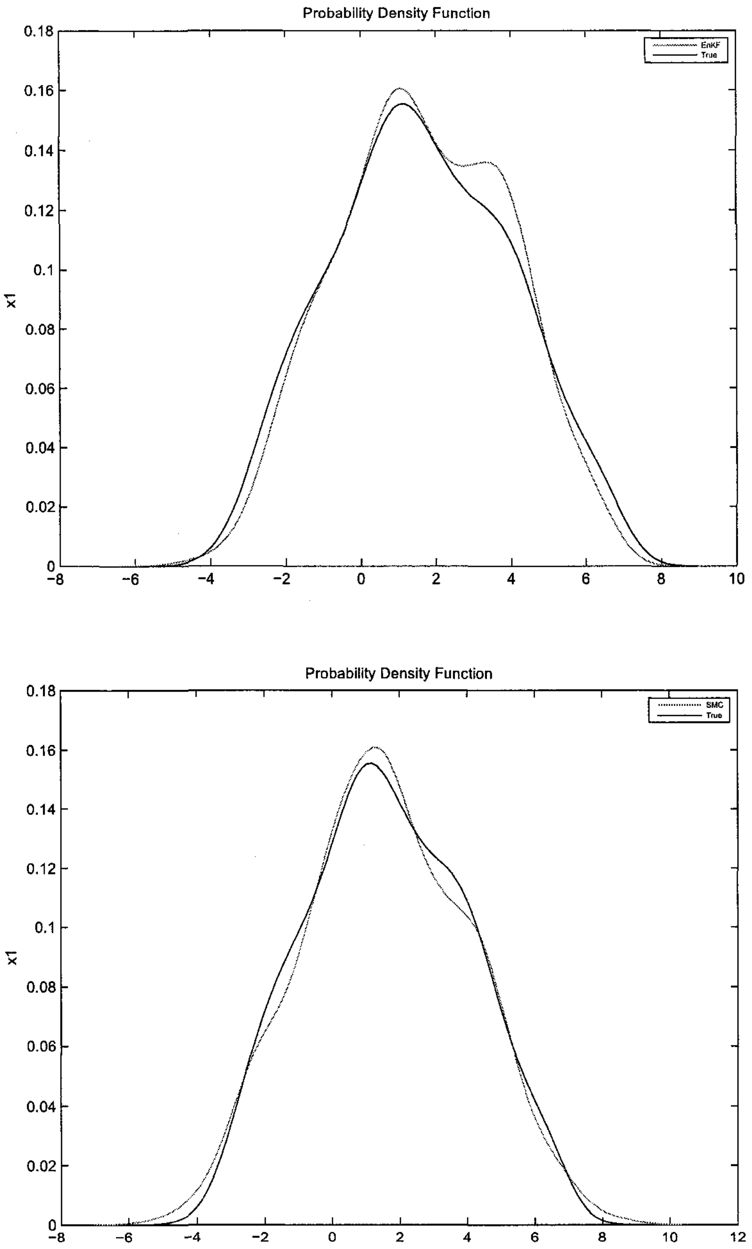


Figure 4.2: Probability density function of $X1$ of Lorenz 1996 model ($F = 5$) for EnKF and SMC methods with filter size of 250

4.2 Highly Chaotic $F = 8$

When $F = 8$, the dynamics become highly chaotic. In Fig 4.3, both the SMC methods and the EnKF method can track the phase transitions and also in reproducing the correct amplitudes of the reference solution reasonably well, not as good as in the case with $F = 5$ though. If we still divide the whole period into two. There is not much difference between the two halves in performance. When degree of chaos increases, the difficulty to track the true resolution increases. From the error variance variation with time, we also can see the strong error growth at those phase transition locations.

For the EnKF method, at more than 1/3 of the whole time period, the error variance is greater than 0.006, which means filter estimates deviate from true solution quite frequently. These locations are distributed in the whole assimilation period. For SMC methods, at more than 1/3 of the whole assimilation period, the error variance is greater than 0.006. Those locations also exist in the whole assimilation period. Compared to Fig 4.1, the data assimilation performance for both methods is worse than that in case $F = 5$. The noisy level of $RMSE$ is also greater than that in case $F = 5$. Despite these divergences, both methods recover quickly and track the reference solution again in general.

Table 4.4 indicates the CPU computation time and the $RMSE$ for both methods in the case. The CPU computation time is 21.968 s for SMC methods, while it is 148.003 s for EnKF method. The CPU computation time for the SMC methods increase significantly from 15.809 s to 21.968 s with the degree of chaos from $F = 5$ to $F = 8$. The EnKF method takes almost 9 times longer than the SMC methods. The $RMSE$ is 1.8116, 1.8128, and 1.8967 for $X1$, $X20$, and $X30$ for the SMC methods, while it is 1.6783, 1.6406, and 1.8820 for $X1$, $X20$, and $X30$ for the EnKF method. The EnKF method outperforms the SMC methods.

From Figure 4.4, the probability density function of $X1$ is still weakly non-Gaussian. Both methods can assimilate the probability density functions quite well in general, however, the EnKF method cannot reach the exact peak of the probability density function, while the SMC methods produced two false peaks of the probability density function.

Table 4.5 shows the statistics of the probability density function of $X1$ of Lorenz 1996 model. The mean, the standard deviation, and the coefficient of skewness for the SMC methods are closer to the true state than those of the EnKF method. However, the coefficient of kurtosis is -0.598373, -0.502227, and -0.661721, which shows the EnKF method assimilates better. Besides, the first and third quartiles and ranges estimate in the EnKF method are better than those from the SMC methods except the second quartile (median).

Table 4.4: Computation Time and RMSE of Lorenz 1996 ($F = 8$)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	21.968	148.003
RMSE (X1)	1.8116	1.6783
RMSE (X20)	1.8128	1.6406
RMSE (X30)	1.8967	1.8820

Table 4.5: Statistics of PDF of $X1$ of Lorenz 1996 ($F = 8$)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	1.6262	1.6123	1.6853	0.0138	-0.0591
Standard deviation	2.3589	2.4508	2.2390	-0.0918	0.1198
Coefficient of skewness	0.0602	0.0438	-0.0397	0.0164	0.1000
Coefficient of kurtosis	-0.6937	-0.4271	-0.6882	-0.2666	-0.0055
First Quartile	-0.8161	-0.7196	-0.7972	-0.0965	-0.0189
Second Quartile	1.8568	1.8654	1.9207	-0.0086	-0.0639
Third Quartile	4.5321	4.4760	4.5672	0.0561	-0.0351
Range	19.8032	20.4565	19.3982	-0.6533	0.4050

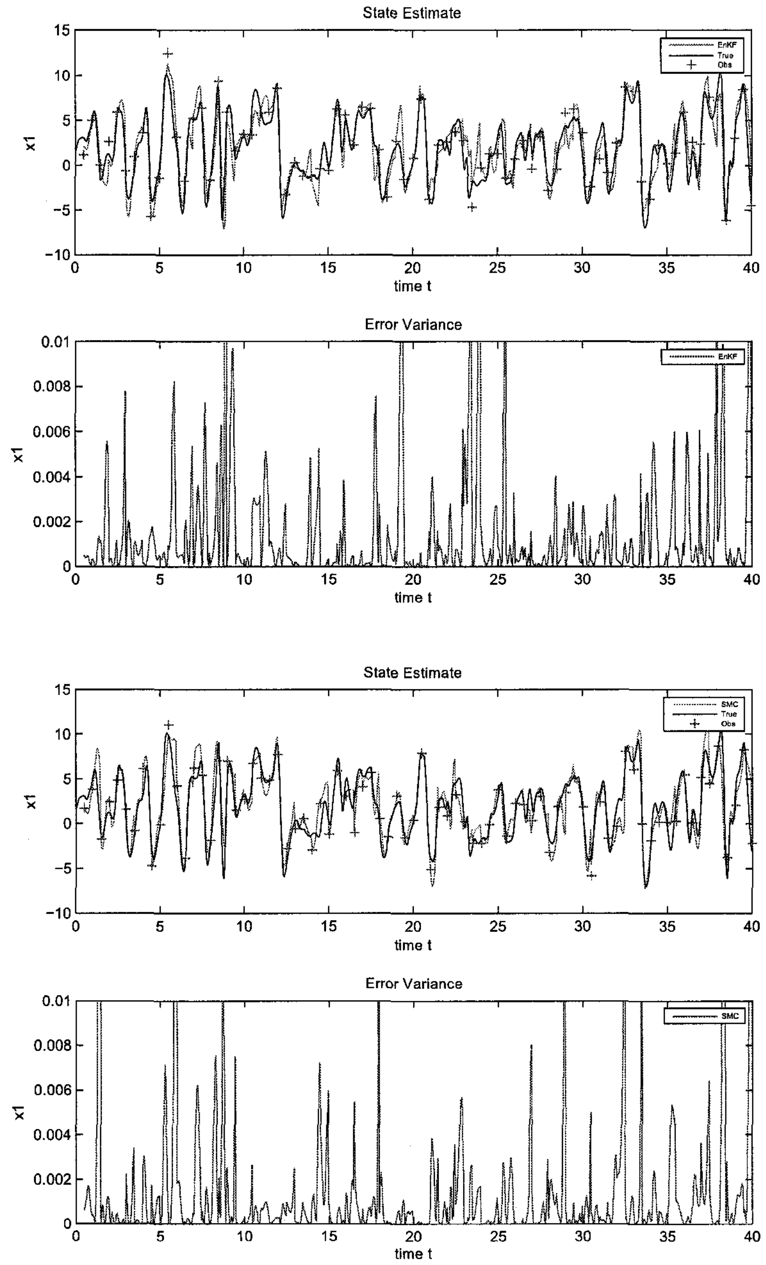


Figure 4.3: State estimate and error variance of $X1$ of Lorenz 1996 model ($F = 8$) for EnKF and SMC methods with filter size of 250

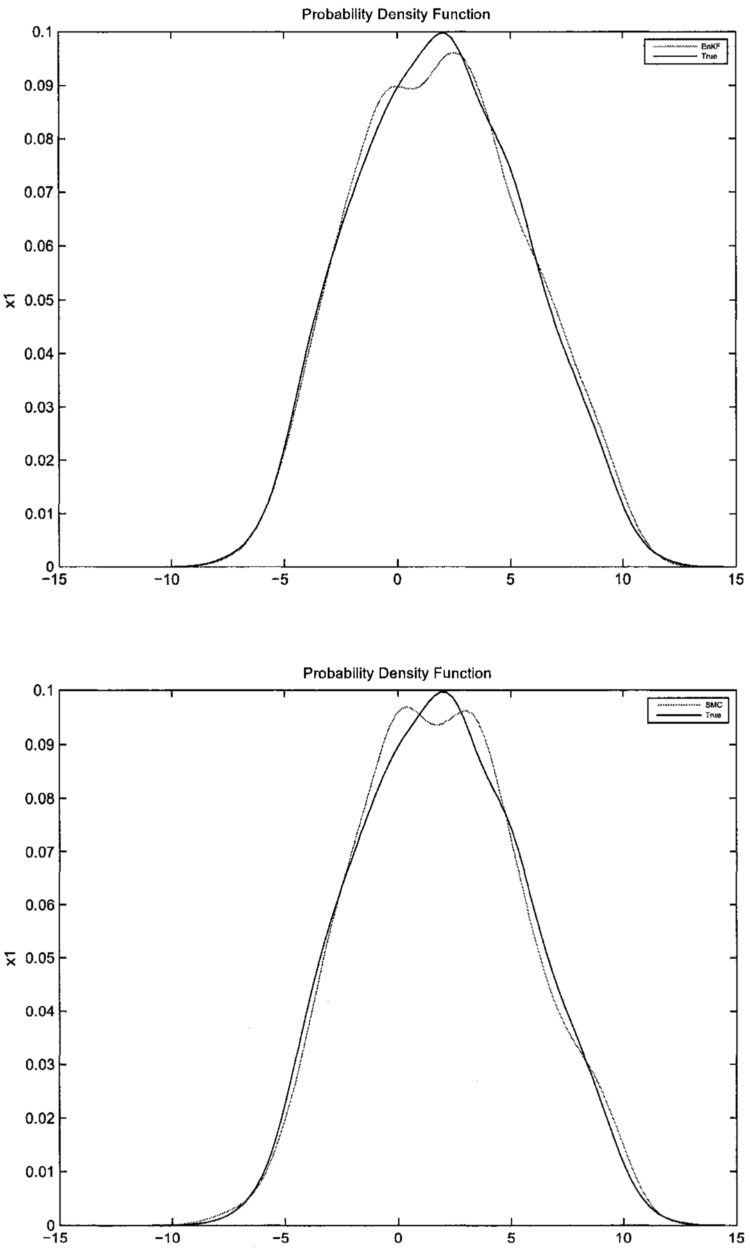


Figure 4.4: Probability density function of $X1$ of Lorenz 1996 model ($F = 8$) for EnKF and SMC methods with filter size of 250

4.3 Fully Turbulent $F = 16$

If we continue to increase the external forcing F, when $F = 16$, the dynamics become fully turbulent. In Fig 4.5, both the SMC methods and the EnKF method can track the phase transitions and also reproduce the correct amplitudes of the reference solution. Obviously it is worse than the first two cases with $F = 5$ and $F = 8$. Since the degree of chaos increases, the whole dynamic process becomes highly unstable and fully turbulent, which increases the difficulties for the data assimilation system. The locations where the filter estimates start to diverge from the reference solution become more frequent than in the first two cases. These locations appear also in the whole assimilation period. From the error variance plot with time, we can see the strong error growth at those phase transition locations. The noisy level of $RMSE$ increases significantly. We cannot use 0.006 as a standard any more. In this case, we choose 0.01 as the standard. For the EnKF method, at more than 1/2 of the whole time period, the error variance is greater than 0.01, which means filter estimates deviate from the true solution significantly and frequently. These locations are in the whole assimilation period. For SMC methods, at more than 1/2 of the assimilation time period, the error variance is greater than 0.01.

Table 4.6 indicates the CPU computation time and the $RMSE$ for both methods in this case. The CPU computation time is 33.493 s for SMC methods, while it is 161.101 s for EnKF method. The EnKF method takes almost 4 times longer than the SMC methods. One interesting feature is that the CPU computation time used by the SMC methods increased significantly with the degree of chaotic nature, while the EnKF method does not. This may indicate that the SMC methods depend on model chaotic nature to some extent. The $RMSE$ is 3.9114, 3.9858, and 3.5495 for $X1$, $X20$, and $X30$ for SMC methods, while it is 3.3883, 3.7520, and 3.7619 for $X1$, $X20$, and $X30$ for EnKF method. Both $RMSE$ from two methods are quite similar.

From Figure 4.6, we can see clearly that the probability density function is non-Gaussian, the same as the previous two cases. Both methods can assimilate it quite well, but, they both have some difficulties to reach the exact peaks of the probability density function.

Table 4.7 shows the statistics of the probability density function of $X1$ of Lorenz 1996 model. The difference of the mean between true state and estimate for the SMC methods are much larger than that of the EnKF method as well as the coefficients of skewness and kurtosis. While the standard deviation of the estimate for the SMC methods are closer to that of the true state than the EnKF. Besides, the first and third quartiles and ranges estimate in the EnKF method are better than those from the SMC methods except the median. The range estimate in the SMC method is almost 10 times larger than that in the EnKF estimate, which means the SMC methods generate some extreme values in the assimilated process, more unstable than the EnKF.

Table 4.6: Computation Time and RMSE of Lorenz 1996 ($F = 16$)

Assimilation Method	SMC(250 particles)	EnKF(250 ensembles)
Time (In Seconds)	33.493	161.101
RMSE (X1)	3.9114	3.3883
RMSE (X20)	3.9858	3.7520
RMSE (X30)	3.5495	3.7619

Table 4.7: Statistics of PDF of $X1$ of Lorenz 1996 ($F = 16$)

Statistics	True PDF	SMC(250 particles)	EnKF(250 ensembles)	True - SMC	True - EnKF
Mean	3.5728	3.2766	3.5611	0.2961	0.0116
Standard deviation	6.2902	6.2494	6.0941	0.0408	0.1961
Coefficient of skewness	0.0669	-0.0039	0.0651	0.0709	0.0018
Coefficient of kurtosis	-0.3906	-0.4688	-0.4474	0.0782	0.0568
First Quartile	-0.7733	-1.1512	-0.9289	0.3779	0.1556
Second Quartile	3.5129	3.0752	3.5134	0.4377	-0.0005
Third Quartile	7.9025	7.7888	8.0694	0.1137	-0.1669
Range	36.4173	40.1501	35.8220	-3.7328	0.5953

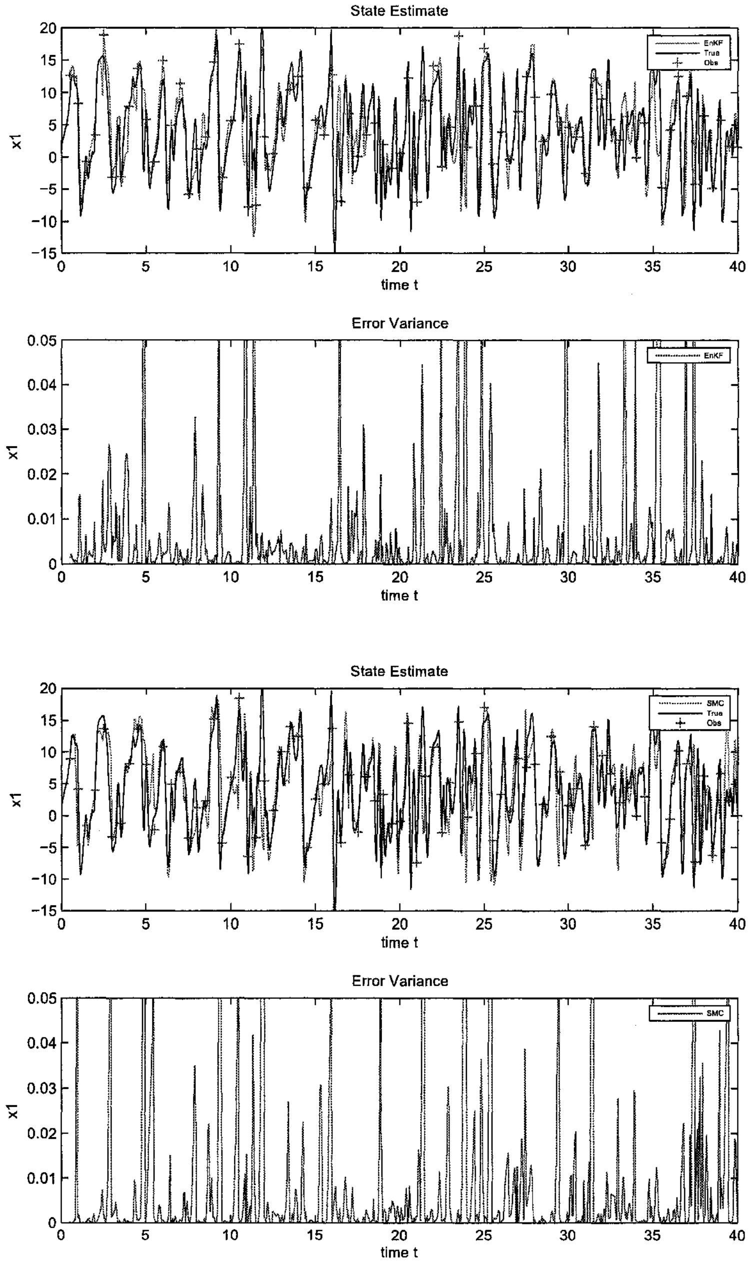


Figure 4.5: State estimate and error variance of $X1$ of Lorenz 1996 model ($F = 16$) for EnKF and SMC methods with filter size of 250

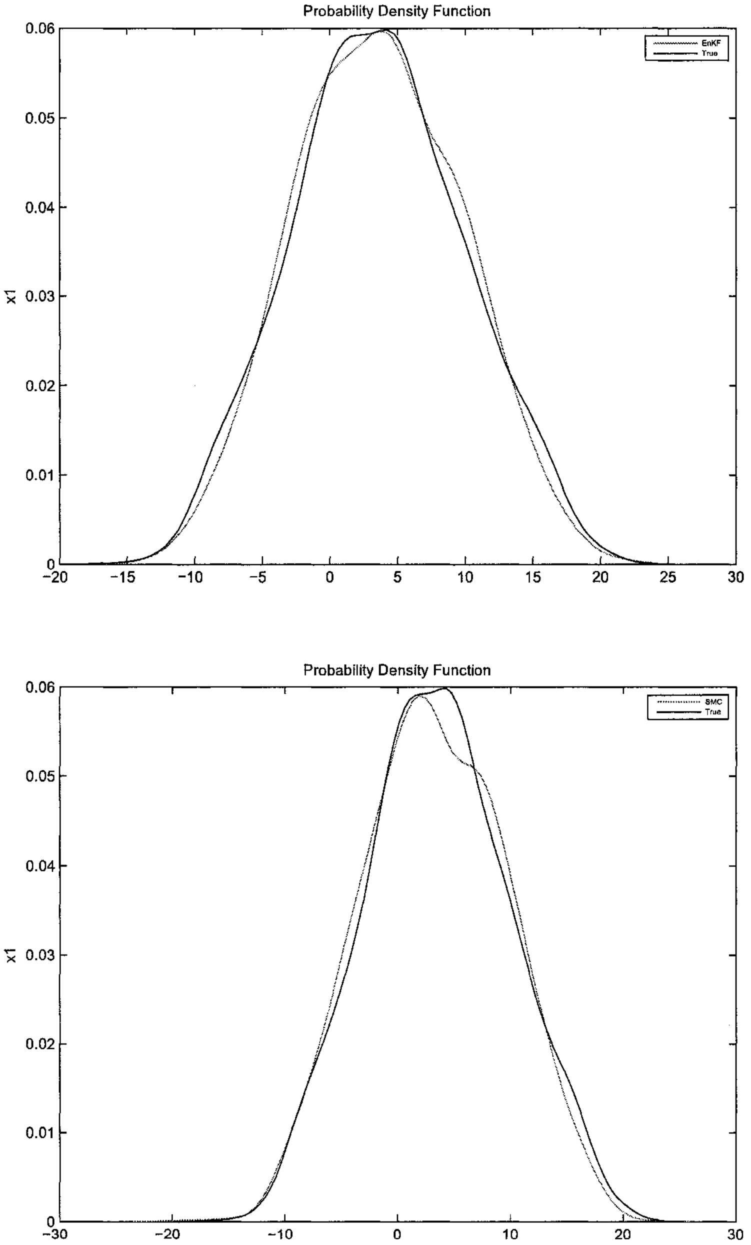


Figure 4.6: Probability density function of $X1$ of Lorenz 1996 model ($F = 16$) for EnKF and SMC methods with filter size of 250

4.4 Filter Size Comparison

In the EnKF method, the error statistics such as mean and variance (covariance) are represented using the model state ensemble. In the SMC methods, the error statistics of the probability density function are estimated using a set of sample particles. In both methods, the larger the ensemble size, the better estimate of the error statistics. As the ensemble size approaches infinity, the estimate of error statistics reaches the optimal estimate.

In practice, it is impossible to employ an ensemble of infinite model members or sample particles to perform data assimilation. In realistic applications, especially in the atmospheric and oceanic fields, the ensemble size varies from 10 to 1000, because the computational cost limits the ensemble size. Since the ensemble size is limited, the estimate based on limited ensembles is not optimal, it is suboptimal. In this section, we compare the different ensemble size of two data assimilation methods. The filter size varies from 5, 10, 25, 50, 75, 100, 250, 500, 750 and 1000.

Figure 4.7 shows the effect of ensemble size on data assimilation in the Lorenz 1963 model. Figures 4.8 and 4.9 show the effect of ensemble size on data assimilation in the Loren 1996 model.

It is clear that $N = 100$ is the critical point of the ensemble size in both the EnKF method and the SMC methods. The $RMSE$ is quite large and oscillates when the ensemble size N is smaller than 100, while the $RMSE$ is quite stable when the ensemble size N is larger than 100. It indicates that in Lorenz models, the ensemble size of 100 is sufficient to achieve reasonably good estimates.

However, in Fig 4.7, the ensemble size of the SMC methods decreases more quickly than that of the EnKF method when the ensemble size is smaller than 100, and it still decreases slowly when the ensemble size is larger than 100, while EnKF does not. In Fig 4.8 and 4.9, the ensemble size of the EnKF method decreases more quickly than that of the SMC methods when the size is smaller than 100. After N greater than 100, both of them are stable.

However, the number of particles needed to perform a successful Sequential Monte Carlo methods increases exponentially with the size of the assimilated dynamic system (Snyder *et al.*, 2008). It is not practical to implement the SMC methods directly to atmospheric and oceanic models at present, since the model has 10^7 degrees of freedom.

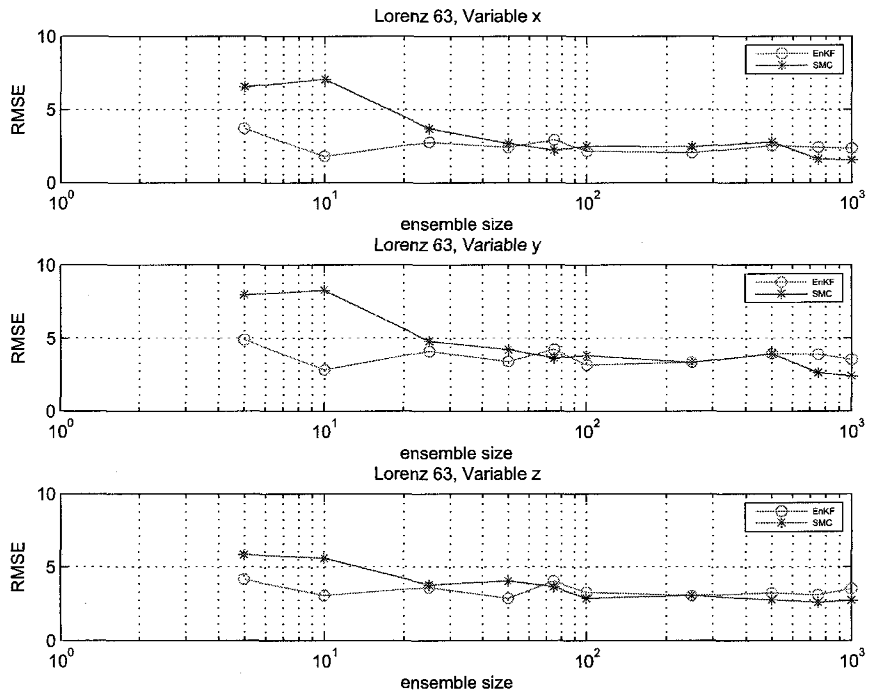


Figure 4.7: Filter size of Lorenz 1963 model for EnKF and SMC methods

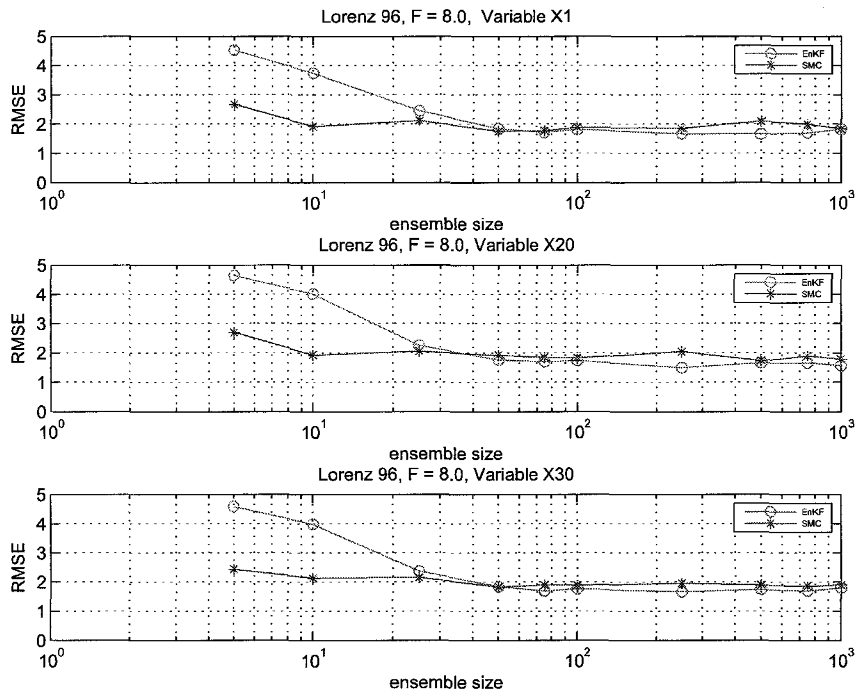


Figure 4.8: Filter size of Lorenz 1996 ($F = 8$) model for EnKF and SMC methods

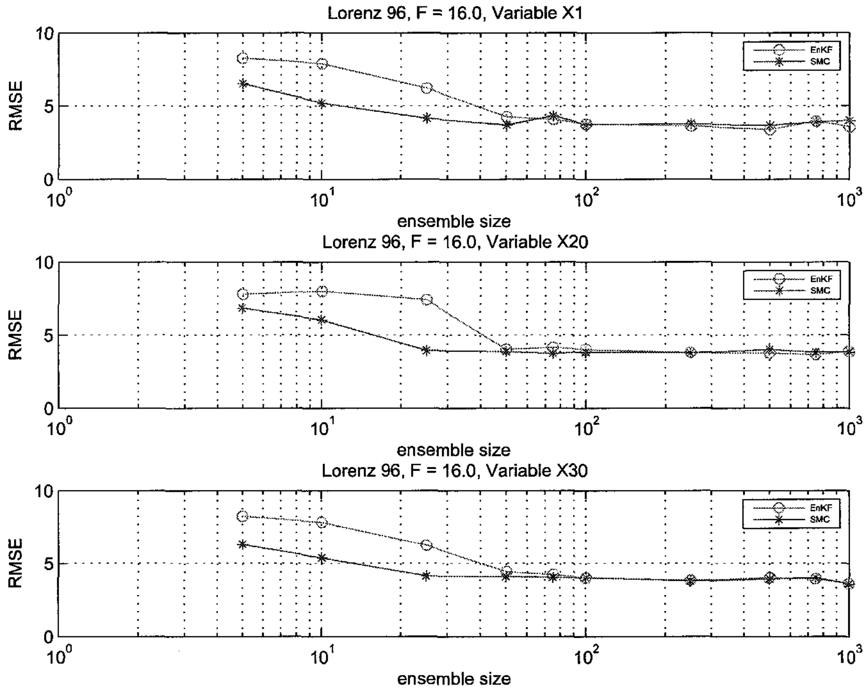


Figure 4.9: Filter size of Lorenz 1996 ($F = 16$) model for EnKF and SMC methods

Chapter 5

Discussions and Conclusions

Although significant progress has been made in the data assimilation field, it is still difficult to deal with the nonlinear and non-Gaussian state estimation problems, which cannot be resolved with the traditional data assimilation methods.

The Sequential Monte Carlo (SMC) methods are among the latest innovations that attempt to bridge the existing gap between the Gaussian dynamics estimation and the non-Gaussian dynamics estimation in the data assimilation process. It has been shown that they perform quite well in the complex practical scenarios.

In the Kalman Filter framework, nonlinearity and non-Gaussianity state estimation problems cannot be solved theoretically; they can only employ the Gaussian probability density function to characterize the non-Gaussian probability density function. Therefore, to tackle this estimation problem of the nonlinear, non-Gaussian dynamic system, SMC methods directly approximate the probability density function (PDF) associated with the dynamic system with finite samples, which is a powerful tool to characterize the uncertainty of the dynamic system instead of only mean and covariance of the system. Different order statistical moments such as mean and variance can be calculated directly from the probability density function.

In this thesis, the SMC methods were used to perform data assimilation in strongly nonlinear dynamic systems, the Lorenz 1963 and 1996 models. Comparison in the same scenarios is made to the EnKF data assimilation method. In the experiment design with the three variables Lorenz 1963 model, we choose

4 scenarios: observation intervals with 0.50 and 0.25, initial non-Gaussian error probability density functions of *Beta* distribution and *Gamma* distribution. In the experiment design with the 40 variables Lorenz 1996 model, we use 3 scenarios: the weakly chaotic dynamics with external forcing $F = 5$, the highly chaotic dynamics with external forcing $F = 8$, and the fully turbulent dynamics with external forcing $F = 16$. Both Lorenz models are relatively low dimensional simplified dynamic models, but they are highly nonlinear dynamics of stochastic nature. Different model parameters represent different situations in the realistic world.

However, in the experiments with Lorenz 1963 and 1996 models as test beds, the SMC methods perform almost as well as the EnKF method, which does not outperform the EnKF method as it is in the theoretical aspect. The reasons may be: 1) the non-Gaussianity of all the probability density functions either in the Lorenz 1963 dynamics or the Lorenz 1996 dynamics are not significantly strong. Those probability density functions are not obviously bimodal or multimodal, though the coefficients of skewness and kurtosis exist. Since the Gaussian probability density function is completely characterized by the first two moments, it is clear that one can obtain the probability density function of a Gaussian process from its mean and covariance information. That explains why the EnKF method generates good results as well as the SMC methods. 2) The criteria $RMSE$ and assimilated probability density function are calculated only from mean values of the true resolution and the assimilated resolution. There are no criteria considering the higher order moments of the error statistics, which is the advantage of the SMC methods.

Since all the data assimilation work are based on Bayesian statistics, the sensitivity analysis of different prior probability density function could be done in the future work to generate the optimal prior probability density function for the data assimilation system.

Although Sequential Monte Carlo methods are suitable for the most general case: nonlinear, non-Gaussian dynamics, for linear, Gaussian dynamics, Kalman Filter will still be the first approach for its easy implementation; for linear or nonlinear, weakly non-Gaussian dynamics as in our experiments, Extended Kalman Filter and Ensemble Kalman Filter could be employed to achieve reasonably good estimate; for highly nonlinear, strongly non-Gaussian dynamics, Sequential Monte Carlo methods will the best choice to fully capture the non-Gaussianity of the dynamics. In the Meantime, the choice of the assimilation method is limited by the computation power.

One interesting feature found in our experiments is that the computational cost of the SMC methods is much cheaper than the EnKF method. This seems inconsistent with the perception that the SMC methods are very expensive computationally. This is mainly because of relatively low dimensionality of the dynamic systems used in this study. As the dimensionality increases, the ensemble size required for SMC methods increases exponentially (Snyder *et al.*, 2008).

Within the theoretical insight, Bengtsson *et al.* (2008) point out that Sequential Monte Carlo (SMC) methods may fail in large scale dynamic systems. Their simulations suggest that the convergence to unity occurs unless the ensemble grows super-exponentially in the system dimension. At present there is no SMC application in realistic atmospheric and oceanic models because of the high dimension of dynamic models. This is an obstacle to high-dimensional SMC data assimilation. According to Snyder *et al.* (2008), Gaussian errors, simulations indicate that the required ensemble size scales exponentially with the state dimension. In his example, the particle filter requires at least 10^{11} members when applied to a 200-dimensional state for the posterior mean from the particle filter to have an expected error smaller than either the prior or the observations.

However, in some cases, the system model has some substructure which can be tractable and analytically marginalized out. The advantage of this strategy is that it can drastically reduce the size of the space over which we need to sample and reduce the filter size. Marginalizing out some of the variables is a process which is called Rao-Blackwellisation, because it is related to the Rao-Blackwell formula: see (Casella & Robert, 1996) for a general discussion.

In this thesis, the properties and capabilities of the SMC methods is investigated and compared to the EnKF method using the low dimensional, highly nonlinear dynamic systems, the Lorenz 1963 and 1996 models. Despite the interesting fact that the EnKF method performs as well as the SMC methods in the highly nonlinear dynamics, the SMC methods have theoretical advantages and potential practical significance, which is helpful when we design data assimilation systems for nonlinear, non-Gaussian realistic models.

Appendix A

FORTRAN Code for the EnKF Method

The EnKF algorithm implemented in this thesis is explained in Evensen (2003). For the detailed description, please refer to Evensen (2003). The following FORTRAN code provides a detailed implementation of the EnKF analysis scheme. It assumes access to the Numerical Algorithms Group (NAG) Numerical Libraries, which specializes in the provision of software for the solution of mathematical, statistical and data mining problems.

```

subroutine EnKF(R, OBS, PCL, xhat)

   implicit none

   integer :: ndim          ! dimension of model state
   integer :: nrens         ! number of ensemble members
   integer :: nrobs         ! number of observations

   integer :: lat, lon              ! number of Latitude, Longitude
integer :: lat_obs , lon_obs   ! number of Latitude , Longitude
integer :: ndim_T              ! grid of the whole domain

parameter (lat=1, lon=3,lat_obs=1, lon_obs=3)
parameter (ndim=lat*lon , nrens=250, nrobs=1)

real      :: PCL(nrens , ndim)       ! input ensemble matrix
real      :: OBS(ndim)               ! input obs matrix
real      :: Xhat(ndim)              ! output analysis


real      :: A(ndim , nrens)         ! Ensemble matrix
real      :: X_A(ndim ,1)            ! Analysis matrix
real      :: X_F(ndim ,1)            ! forecast matrix
real      :: Y(ndim ,1)              ! observations matrix
real      :: X_M(ndim ,1)            ! Ensemble mean matrix

real      :: X_A2(ndim , nrens)      ! updated all member analysis
    matrix
real      :: Y_P2(ndim , nrens)      ! perturbed observations
    matrix

real      :: K(ndim , ndim)          ! Kalman Gain matrix
real      :: D(ndim ,1)              ! Innovation matrix

real      :: P_A(ndim , ndim)        ! analysis error covariance
    matrix
real      :: P_F(ndim , ndim)        ! forecast error covariance
    matrix
real      :: P_R(ndim , ndim)        ! observations error
```

```
        covariance matrix

    real      :: H(ndim,ndim)              ! observation operator matrix
    real      :: I(ndim,ndim)              ! identity matrix

! Local variables

    real, allocatable, dimension(:, :)  :: X1, X2, X3, X4, X5, X6, &
                                           X7, X8, X9, X11

    integer    ::     NMAX, LDA, LWORK, INFO, N
    parameter        (NMAX=lat*lon, LDA=NMAX, LWORK=64*NMAX)
    real,      allocatable, dimension(:)   :: WORK
    integer,   allocatable, dimension(:)   :: IPIV

    external   ::     F07ADF, F07AJF, F06YAF

    integer    ::     t, j, kappa, u
    real       ::     alpha=1.0, beta=0.0, theta=1.0/(nrens-1)

    real, allocatable, dimension(:, :)  :: X_F3

    character :: date, time, zone
    integer, dimension(8) :: values

    real :: start, finish, R

    integer   :: N02, N03
    parameter  (N02 = ndim*nrens, N03 = ndim*nrens)
```

```
   integer   :: IFAIL, IGEN

   real      :: X02(N02), X03(N03)
   integer   :: ISEED(4)

   external     G05KCF, G05LAF

! ************************************************************
! Assimilation Cycle Starts.
! ************************************************************

   do t = 1, ndim
      do j = 1, nrens
         A(t, j) = PCL(j, t)
      end do
   end do

!   Perturbed Observation
!
!   Initialize the seed to a un-repeatable sequence
         ISEED(1) = 1762543
         ISEED(2) = 9324783
         ISEED(3) = 42344
         ISEED(4) = 742355
!   IGEN identifies the stream
         IGEN = 1
         call G05KCF(IGEN, ISEED)
!
         IFAIL = 0
         call G05LAF(0.0e0, R, N02, X02, IGEN, ISEED, IFAIL)
```

```
        Y_P2 =  reshape(X02, (/ndim, nrens/))

! **********************************************
  do u = 1, nrens
! **********************************************

  do t = 1, ndim
      X_F(t, 1) = A(t, u)
  end do

  do t = 1, ndim
    Y(t, 1) = OBS(t) + Y_P2(t, u)
  end do

! generate the identical matrix
! H(ndim,ndim)

  do t=1,ndim
     do j=1,ndim
         if (t==j) then
             H(t,j)=1.0
         else
             H(t,j)=0.0
         end if
     end do
  end do

! **********************************************************
! Compute Background error covariance P_F
! **********************************************************

!  A(ndim,nrens)
!  X_M(ndim,1)  ensemble  mean

    do t=1,ndim
        X_M(t,1)=sum(A(t,:))/nrens
    end do

! background  error  covariance
! X_F(ndim,ndim)

! error
  do t=1, nrens

      allocate (X_F3(ndim, 1))
      X_F3(1:ndim, 1) = A(1:ndim, t) - X_M(1:ndim, 1)

! error covariance
! X_F3(ndim,1)
! X_F3(ndim,1)
! P_F(ndim,ndim)

      allocate (X11(ndim,ndim))
      call F06YAF('n', 't', ndim, ndim, 1, &
                     alpha, X_F3, ndim, &
                            X_F3, ndim, &
                     beta, X11, ndim)

      P_F=X11 + P_F
      deallocate (X_F3)
      deallocate (X11)

  end do

  P_F=P_F*theta

! ***************************************************************
! step 1. compute innovation d
! ***************************************************************

!  1) compute X1=H^X_F
! H(ndim,ndim)
! X_F(ndim,1)
! X1(ndim,1)

    allocate (X1(ndim,1))
    call F06YAF('n', 'n', ndim, 1, ndim, &
                  alpha, H, ndim, &
                        X_F, ndim, &
                  beta, X1, ndim)

!  2)compute d=y0-H^X_F=y0-X1
!  Y(ndim,1)
!  X1(ndim,1)
!  D(ndim,1)

    D=Y-X1
    deallocate (X1)
```

```
! *****************************************************************
! step 2. compute gain matrix K
! *****************************************************************

! 1) compute Pf^H'
! P_F(ndim,ndim)
! H(ndim,ndim)
! X2(ndim,ndim)

  allocate (X2(ndim,ndim))
  call F06YAF('n', 't', ndim, ndim, ndim, &
              alpha, P_F, ndim, &
                  H, ndim, &
              beta, X2, ndim)

! 2) compute H^P_F
! H(ndim,ndim)
! P_F(ndim,ndim)
! X3(ndim,ndim)

  allocate (X3(ndim,ndim))
  call F06YAF('n', 'n', ndim, ndim, ndim, &
              alpha, H, ndim, &
                  P_F, ndim, &
              beta, X3, ndim )  ! (about 20 min)

! 3) compute X3^H'
! X3(ndim,ndim)
! H(ndim,ndim)
! X4(ndim,ndim)

    allocate (X4(ndim,ndim))
    call F06YAF('n', 't', ndim, ndim, ndim, &
                    alpha, X3, ndim, &
                           H, ndim, &
                    beta, X4, ndim )
    deallocate (X3)

! 4) compute R+X4
! P_R is the diagonal matrix

    do t=1,ndim
        do j=1,ndim
            if (t==j) then
                P_R(t,j)=R
            else
                P_R(t,j)=0.0
            end if
        end do
    end do

! X5(ndim,ndim)

    allocate (X5(ndim,ndim))
    X5=P_R+X4
    deallocate (X4)

! 5) compute inverse of X5

```

```
   allocate (WORK(LWORK))
   allocate (IPIV(NMAX))

!  Factorize X5
  N = ndim
       call F07ADF(N,N,X5,LDA,IPIV,INFO)
       if (INFO.EQ.0) then
!  compute inverse of X5
           call F07AJF(N,X5,LDA,IPIV,WORK,LWORK,INFO)
       endif

   deallocate (WORK)
   deallocate (IPIV)


! 6) compute gain K=
! X2(ndim,ndim)
! X5_inverse=(ndim,ndim)
! K(ndim,ndim)

   call F06YAF('n', 'n', ndim, ndim, ndim, &
               alpha, X2, ndim, &
                    X5, ndim, &
               beta, K, ndim)
   deallocate (X2)
   deallocate (X5)

! *************************************************************
! Step 3. compute X_a
! *************************************************************

! 1) compute K^d
! K(ndim,ndim)
! D(ndim,1)
! X7(ndim,1)

      allocate(X7(ndim,1))
      call F06YAF('n', 'n', ndim, 1, ndim, &
                        alpha, K, ndim, &
                               D, ndim,  &
                        beta, X7,ndim)

! 2) compute X_a
! X_F(ndim,1)
! X_A(ndim,1)

    X_A=X_F+X7
    deallocate (X7)

    do t= 1, ndim
        X_A2(t, u) = X_A(t, 1)
    end do

! ********************************************************
! Step 4. compute P_A
! ********************************************************

! 1) compute K^H
! K(ndim,ndim)
! H(ndim,ndim)
! X8(ndim,ndim)
```

```
    allocate (X8(ndim,ndim))
    call F06YAF('n', 'n', ndim, ndim, ndim, &
                alpha, K, ndim, &
                     H, ndim,  &
                beta, X8, ndim)


! 2) compute I-K^H
! I(ndim,ndim)
! X9(ndim,ndim)

    allocate (X9(ndim,ndim))
    X9=I-X8
    deallocate (X8)

! 3) compute P_a
! P_A(ndim,ndim)
! P_F(ndim,ndim)
! X9(ndim,ndim)

    call F06YAF('n', 'n', ndim, ndim, ndim, &
                alpha, X9, ndim, &
                     P_F, ndim, &
                 beta, P_A, ndim)
    deallocate (X9)

! *********************************************
    end do
! *********************************************
```

```
    do t = 1, ndim
        Xhat(t) = sum(X_A2(t, :))/nrens
    end do

    do t = 1, ndim
        do j = 1, nrens
            PCL(j, t) = X_A2(t, j)
        end do
    end do

    P_F = 0.0
    P_A = 0.0

    return
end subroutine EnKF
```

Appendix B

FORTRAN Code for the SMC Methods

The SMC algorithm implemented in this thesis is explained in Gordon *et al.* (1993). For the detailed description, please refer to Gordon *et al.* (1993). The following FORTRAN code provides a detailed implementation of the SMC analysis scheme.

```

      subroutine Particle_Filter (R, y, x, xhat)

      integer :: i, j, M
      parameter (M = 250)

      real, dimension (M, 3) :: x, e, q_w, qn, ind, temp01
      real, dimension (3) :: y, PROB00, xhat, q_sum, temp11, prob
      real, parameter  :: pi = 3.1415926
      real :: R

! step 1.
!  Calculate difference between Observations and model forecasts

      do i = 1, M
         e(i, :) = y(:) - x(i, :)
      end do

! step 2.
! Calculate weights from Gaussian Distribution
      do i = 1, M
         call Std_Normal(e(i, :), q_w(i, :), R)
      end do

! step 3.
! summation of all the weights
      do i = 1, 3
         q_sum(i) = sum (q_w(:,i))
      end do

! step 4.
! Normalize importance weights
      do j = 1, 3
         do i = 1, M
            q_w(i, j) = q_w(i, j) / q_sum(j)
         end do
      end do

! step 5.
! Get particle * weight
      do j = 1, 3
         do i = 1, M
```

```
            qn(i, j) = q_w(i, j) * x(i, j)
         end do
      end do

! step 6.
! Get analysis
      do i = 1, 3
         xhat(i) = sum (qn(:,i))
      end do

! step 7 ** Resampling **
      call resampling (q_w, ind)

      do j = 1, 3
         do i = 1, M
            x(i, j) = x(ind(i, j), j)
         end do
      end do

      return
      end subroutine Particle_Filter

! ***************************************************************
!     Resampling Process

      subroutine resampling (q_w, ind)
        implicit none

        integer :: M
        parameter (M = 250)
```

```
real, dimension (M,3) :: q_w, ind, qc, xxx, yyy, u, temp01
real, dimension (3)    :: ssum, ran01
integer :: i, j, k, n

ssum = 0.0
do j = 1, 3
   do i = 1, M
      ssum(j) = ssum(j) + q_w(i, j)
      qc(i, j) = ssum(j)
   end do
end do

call random_seed ()
call random_number (ran01)

do j = 1, 3
   do i = 1, M
      u(i, j) = (i - 1 + ran01(j))/M
   end do
end do

do n = 1, 3
   do j = 1, M
      k = 1
      do while (qc(k, n) < u(j, n))
          k = k + 1
      end do
   temp01(j, n) = k
   end do
```

```
         end do
         ind = temp01

         return
       end subroutine resampling

! ***************************************************************
!                           END
! ***************************************************************
```

References

ANDERSON, B.D.O. & MOORE, J.B. (1979). *Optimal Filtering*. Information and System Science Series, Prentice Hall, Englewood Cliffs, NJ, USA. 8

ARULAMPALAM, M.S., MASKELL, S., GORDON, N. & CLAPP, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, **50(2)**, 174–188. 8, 13

BENGTSSON, T., BICKEL, P. & LI, B. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. *Probability and Statistics: Essays in Honor of David A. Freedman, IMS Collections*, **2**, 316–334. 65

BURGERS, G., VAN LEEUWEN, P.J. & EVENSEN, G. (1998). Analysis scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, **126**, 1719–1724. 3, 7, 9

CASELLA, G. & ROBERT, C.P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, **83(1)**, 81–94. 65

COHN, S.E. (1997). An introduction to estimation theory. *Journal of the Meteorological Society of Japan*, **75**, 147–178. 6

COURTIER, P., ANDERSSON, E., HECKLEY, W., PAILLEUX, J., VASILJEVIC, D., HAMRUD, M., HOLLINGSWORTH, A., RABIER, F. & FISHER, M. (1998). The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, **124(550)**, 1783–1808. 2, 3

DIMET, F.X.L. & TALAGRAND, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus*, **38A**, 97–100. 2

DOUCET, A., GODSILL, S.J. & ANDRIEU, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10(3)**, 197–208. 8, 13

DOUCET, A., DE FREITAS, N. & GORDON, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, USA. 7, 8

EFRON, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, **7(1)**, 1–26. 12

EVENSEN, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, **99(C5)**, 10143–10162. 3, 9, 15

EVENSEN, G. (1997). Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, **125**, 1342–1354. 23

EVENSEN, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, **53**, 343–367. 3, 7, 29, 67

EVENSEN, G. & VAN LEEUWEN, P.J. (1996). Assimilation of geosat altimeter data for the Agulhas current using the Ensemble Kalman Filter with a quasigeostrophic model. *Monthly Weather Review*, **124**, 85–96. 3, 7

GORDON, N.J., SALMOND, D.J. & SMITH, A.F.M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE proceedings. F, Radar and signal processing*, **140**, 107–113. 4, 8, 13, 15, 80

HOUTEKAMER, P.L. & MITCHELL, H.L. (1998). Data assimilation using an Ensemble Kalman Filter technique. *Monthly Weather Review*, **126**, 796–811. 3, 7

HOUTEKAMER, P.L. & MITCHELL, H.L. (2005). Ensemble Kalman filtering. *Quarterly Journal of the Royal Meteorological Society*, **131**, 3269–3289. 10

HOUTEKAMER, P.L., MITCHELL, H.L., PELLERIN, G., BUEHNER, M., CHARRON, M., SPACEK, L. & HANSEN, B. (2005). Atmospheric data assimilation with an Ensemble Kalman Filter: Results with real observations. *Monthly Weather Review*, **133**, 604–620. 2

ISARD, M. & BLAKE, A. (1998). A smoothing filter for condensation. *Proceedings of 5th European Conference on Computer Vision*, **1**, 767–781. 13

JAZWINSKI, A.H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press, New York, USA. 3, 8

JULIER, S.J. & UHLMANN, J.K. (1996). *A General Method for Approximating Nonlinear Transformations of Probability Distributions*. Technical Report, RRG, Dept. of Engineering Science, University of Oxford. 29

KALMAN, R.E. (1960). A new approach to linear filtering and prediction. *Transactions of the ASME. Series D, Journal of Basic Engineering*, **82**, 35–45. 2, 3, 8

KITAGAWA, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5(1)**, 1–25. 8, 13

KLINKER, E., RABIER, F., KELLY, G. & MAHFOUF, J.F. (2000). The ECMWF operational implementation of four-dimensional variational assimilation - Part III: Experimental results and diagnostics with operational configuration. *Quarterly Journal of the Royal Meteorological Society*, **126**, 1191–1215. 2

LIU, J.S. & CHEN, E. (1998). Sequential Monte Carlo methods for dynamics systems. *Journal of the American Statistical Association*, **93**, 1032–1044. 8, 13

LORENC, A.C. (2003). The potential of the Ensemble Kalman Filter for NWP: A comparison with 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, **129**, 3183–3203. 3, 7

LORENZ, E.N. (1963). Deterministic non-periodic flow. *Journal of the Atmospheric Sciences*, **20**, 130–141. 15

LORENZ, E.N. (1996). Predictability: A problem partially solved. *Proceedings of the Seminar on Predictability, ECMWF*, **1**, 1–18. 41

MAHFOUF, J.F. & RABIER, F. (2000). The ECMWF operational implementation of four-dimensional variational assimilation - Part II: Experimental results with improved physics. *Quarterly Journal of the Royal Meteorological Society*, **126**, 1171–1190. 2

MAJDA, A.J. & HARLIM, J. (2008). Filtering nonlinear dynamical systems with linear stochastic models. *Nonlinearity*, **21**, 1281–1306. 41

MILLER, R.N., GHIL, M. & GAUTHIEZ, F. (1994). Advanced data assimilation in strongly nonlinear dynamical systems. *Journal of the Atmospheric Sciences*, **51**, 1037–1056. 18

PARZEN, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, **33**, 1065–1076. 24

RABIER, F., JARVINEN, H., KLINKER, E., MAHFOUF, J.F. & SIMMONS, A. (2000). The ECMWF operational implementation of four-dimensional variational assimilation - Part I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, **126**, 1143–1170. 2

RUBIN, D.B. (1988). Using the SIR algorithm to simulate posterior distributions. *Bayesian Statistics, Oxford University Press*, **3**, 395–402. 12

SCHMIDT, S.F. (1966). Application of state-space methods to navigation problems. *Advances in Control Systems*, **3**, 293–340. 3, 7, 8

SCHON, T.B. (2006). *Estimation of Nonlinear Dynamic Systems - Theory and Applications*. Linkoping Studies in Science and Technology, Ph.D. Dissertation. 7, 8, 11, 13

SILVERMAN, B.W. (1986). *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability, Chapman and Hall, London. 24

SMITH, A.F.M. & GELFAND, A.E. (1992). Bayesian statistics without tears: A sampling resampling perspective. *American Statistician*, **46(2)**, 84–88. 12

SMITH, G.L., SCHMIDT, S.F. & MCGEE, L.A. (1962). Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. *Technical Report TR, NASA*, **R-135**. 3, 7, 8

SNYDER, C., BENGTSSON, T., BICKEL, P. & ANDERSON, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, **136**, 4629–4640. 58, 65

TALAGRAND, O. (1997). Assimilation of observations, an introduction. *Journal of the Meteorological Society of Japan*, **75**, 191–209. 1

TIPPETT, M.K., ANDERSON, J.L., BISHOP, C.H., HAMILL, T.M. & WHITAKER, J.S. (2003). Ensemble square root filters. *Monthly Weather Review*, **131**, 1485–1490. 3, 7