ISBN 978-90-365-2864-1

TOPICS IN PARTICLE FILTERING AND SMOOTHING

Saikat Saha

TOPICS IN PARTICLE FILTERING AND SMOOTHING

Saikat Saha

Stochastic Volatility Parameter Jumps Gaussian Proposal EMM Smoothed Marginal MAP Sequential Monte Carlo

TOPICS IN PARTICLE FILTERING AND SMOOTHING

The research described in this thesis was undertaken at the Department of Applied Mathematics, in the Faculty EWI, University of Twente, Enschede, The Netherlands. The funding of the research was provided by the THALES Nederland B.V.

Graduation Committee:

prof. dr. ir. A. J. Mouthaan (Chairman), University of Twente, EWI
prof. dr. A. Bagchi (Promotor), University of Twente, EWI
dr. P. K. Mandal (Asst. Promotor), University of Twente, EWI
dr. ir. J. N. Driessen (Referent), Thales Nederland
prof. F. Gustafsson, Linköping University, Sweden
prof. dr. ir. C. H. Slump, University of Twente, EWI
prof. dr. A. A. Stoorvogel, University of Twente, EWI
dr. ir. M. H. Vellekoop, University of Twente, EWI

Printed by Wöhrmann Printing Service, The Netherlands

The book cover is designed by Sangeeta Nath.

The summary in Dutch is done by Prof. Arun Bagchi.

ISBN: 978-90-365-2864-1

TOPICS IN PARTICLE FILTERING AND SMOOTHING

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday 18 September 2009 at 13.15 hours

by

Saikat Saha
born on the 31^{st} of December 1974
Santaldih, India

Dit proefschrift is goedgekeurd door de promotor
Prof. dr. A. Bagchi

en de assistent-promotor,
dr. P. K. Mandal

Contents

Acronyms

CRLB	Cramer-Rao lower bound
CPU	central processing unit
EKF	extended Kalman filter
EMM	exact moment matching
EM	expectation-maximization
EP-VGM	end Point Viterbi-Godsill MAP
GHQ	Gauss-Hermite quadrature
GSF	Gaussian sum filter
IS	importance sampling
JUQ	Julier-Uhlmann quadrature
KLD	Kullback-Leibler divergence
LIN	importance function by linearization (by Doucet *et al.* (2000))
MAP	maximum *a posteriori*
ML	maximum likelihood
MMSE	minimum mean square error
MCMC	Markov chain Monte Carlo
PF	particle filtering
Pf-MAP	particle based filter MAP estimate
SIS	sequential importance sampling
SMC	sequential Monte Carlo
UPF	unscented particle filter (by van der Merwe *et al.* (2000))

Chapter 1

Introduction

1.1 Background

The early works of stochastic filtering started with the problem of statistically extracting the true signal, x_k at time k, when the measurement, y_k of this true signal, x_k, is corrupted by noise, v_k. The main objective considered was to estimate x_k based on the knowledge of y up to time k. The stochastic filtering theory developed significantly in the early 1940's due to the pioneering works of two celebrated mathematicians, Norbert Wiener and A. N. Kolmogorov, who solved the above filtering problem independently. While Kolmogorov's approach was based on the projection theorem on a Hilbert space, Wiener's solution was based on (cross) correlation of signal and observation. Wiener's key assumptions were that (1) the processes are scalar and (2) the signal and noise processes are jointly stationary. Extension to the non stationary case is not so obvious in Wiener's set up, while to extend it to the vector case, the analysis became highly complicated. The real breakthrough came finally through the seminal contribution of R. E. Kalman in the late 50's, popularly known as the Kalman filter. Kalman introduced a model for the signal, which was a dramatic paradigm shift. He came up with the dynamic state space formulation, which can easily cater for non stationary and multi dimensional problems. The Kalman filter found immediate sensational success in the space missions and today the Kalman filter is used practically in every branch of science and engineering. The solution obtained is optimal in the minimum mean square sense and it is recursive in nature, thus allowing on line estimation. However, the Kalman filter is restricted by the assump-

tions of the linear-Gaussian model. On the contrary, nonlinearity and/or non-Gaussianity is abound in the physical world, which motivated the development of different nonlinear filters, such as the extended Kalman filter, Gaussian sum filter and filters based on different numerical quadrature rules. From the early 90's, a class of Monte Carlo simulation based filters, popularly known as Particle filters have appeared to dominate this field due to their flexibility in adapting nonlinearity and/or non-Gaussianity without any ad-hoc assumptions on the models.

1.2 Motivation

Nonlinear non-Gaussian state space models are a quite flexible framework for modeling time series, which often arise in many different practical applications in science and engineering. In this framework, the typical problem of interest is to infer the state of a dynamic system sequentially using a sequence of noisy measurements. Optimal estimate for such models can not generally be obtained in closed form and as a result, many different approximate methods have been proposed as outlined in the previous section. Particle filtering methods were introduced in their modern forms by the pioneering contributions of Gordon *et al.* (1993). Since then, they have become a very popular class of algorithms for solving such problems. The main advantages of this class of algorithm is that they do not make any ad-hoc approximation on the models and the basic algorithm is easy to implement and modular in nature. With these flexibilities, they have been successfully applied for sequential inferences in very complex systems, which were previously thought to be intractable. The success has naturally generated a lot of research interest in this area across different disciplines. Consequently, researchers have started addressing different theoretical issues and many variants of particle filters with successive improved features have been proposed over the times. This is an active area of research and there are many issues that are either not properly addressed or still open.

So far we have discussed the emergence of the particle filter, which provides a numerical solution to the Bayesian filtering problem sequentially using Monte Carlo methods. The complete solution of the filtering problem is given by the posterior distribution of the state given a set of observations. This is in general, analytically intractable, but can be successfully approximated by the particle filters in the form of (weighted) random samples, also known as particles. As we will be describing in Chapter 2, the random

samples are obtained from a different distribution, called the importance distribution (or the importance function), which is ideally supposed to be as close as possible to the posterior. Although particle filtering is an elegant way of approximating the posterior, the efficiency of the method depends heavily on the selection of the importance distribution. Usually in practice, the 'naive' proposal $p(x_k|x_{k-1})$ is used as it is easily available from the model (Gordon *et al.* (1993)). However, by using the naive proposal to explore the state space, a lot of samples are wasted, especially when the measurement is very informative. To make the method more effective importance functions of the form $\pi = p(x_k|x_{k-1}, y_k)$, i.e., the one which incorporates the recent observation as well, are suggested in Liu and Chen (1998) and Doucet *et al.* (2000). It has also been shown by Doucet *et al.* (2000) that the aforementioned importance function is optimal in the sense that the variance of the (unnormalized) importance weights conditional upon the trajectory $x_{0:k-1} \equiv (x_1, x_2, \ldots, x_{k-1})$ and observations $y_{1:k} \equiv (y_1, y_2, \ldots, y_k)$ is minimum. In practice, however, there are two major prohibitive drawbacks for using this type of importance function. Firstly, drawing samples according to $p(x_k|x_{k-1}, y_k)$ is, in general, difficult. Secondly, it is also difficult to get the analytical expression which is needed for the weight update. Naturally, a lot of ongoing research efforts have been devoted on how best to approximate this optimal importance function. This forms the basis of our Chapter 3.

The particle filter provides us with the weighted particle approximation of the posterior. However, for inference purposes, often a point estimate is much more convenient. In principle, one can extract any such point estimate from the posterior. One such popular point estimate is the minimum mean square error (MMSE) estimate, which can be easily obtained as the mean of the weighted particles (Doucet *et al.* (2000)). However, in certain applications like target tracking, where this posterior is often multi modal, the mean may not be always meaningful. For such a scenario, another point estimate, namely the maximum *a posteriori* (MAP) estimate, which corresponds to the maximum of the posterior, may be more relevant. We found that the MAP estimator has not been adequately addressed in the particle filter literature. Usually in practice, the particle with the highest weight is naively taken as the MAP estimate, but recently it has been shown that this is not the true MAP estimator. Subsequent developments for MAP in the filtering and smoothing scenarios form the basis of Chapter 4.

In many situations, the state space model also depends on unknown parameters and their estimation is also of immense interest. The parameter estimation problem for a general state space model using particle based methods has generated a lot of interest over the past few years. In this framework, general solutions for parameter estimation, which are useful for any model, are still limited in performance. This is an active area of research and lot of research activities are going on around this theme because of its enormous practical interest. We have discussed the parameter estimation problem and proposed some new methods for estimating the parameters in Chapter 5.

This area is far from mature, at least in theoretical aspects and there are still many open problems which require further research attention. Possible directions for future research are discussed in Chapter 6.

1.3 Layout and contributions

The thesis is organized into six chapters. The contents of the remaining chapters are briefly summarized as follows.

Chapter 2 This chapter includes a brief overview of particle filtering methods. Starting with the foundation of Bayesian filtering, we present a very basic review of Monte Carlo methods and Importance Sampling. Next, we explain the Sequential Importance Sampling method. We then discuss the problem of degeneracy associated with this method which is followed by the Resampling step to mitigate this problem. At this stage, having discussed all the key ingredients, we explain a generic particle filter algorithm.

Chapter 3 The key question in particle filtering is a suitable choice of the importance function. The optimal importance function as described in Doucet *et al.* (2000) is difficult to obtain in most practical situations. The main contribution of this chapter is determining an approximation of the optimal importance function using the moment matching method. We start this chapter with describing the role of the importance function in particle filtering methods and subsequently define the optimal importance function, which incorporates information from the most recent observation. This optimal importance function is rarely available analytically except in special situations. One such case is the celebrated Heston model with jumps for

(unobserved) volatility of a stock price process, which has been outlined in Appendix B. Subsequently we introduce a new Gaussian importance function (EMM) approximating this optimal function using moment matching methods and outline its implementation. Next, we describe other existing Gaussian importance functions (LIN, GHQ, JUQ, UPF) (Doucet *et al.* (2000); Guo *et al.* (2005); van der Merwe *et al.* (2000)), we outline their differences and compare them numerically. The performances of different importance functions are also compared in terms of Kullback-Leibler divergence. This chapter is based on (Saha *et al.* (2009*b*, 2007, 2006); Aihara *et al.* (2008)).

Chapter 4 The main focus of this chapter is the development of a new smoothed marginal maximum *a posteriori* (MAP) estimator from the available particle cloud representation of the marginal smoother. We start with extracting the MAP estimate from the particle cloud representation of the filter distribution. Usually, the particle with the highest weight is taken as MAP. However, it has been shown recently that this is not the most appropriate MAP estimator (Cappé *et al.* (2007); Driessen and Boers (2008*a*)). Another popular approach is the kernel based method. But this is computationally demanding and needs a bandwidth selection. Here we describe two other alternative approaches to calculate the filter MAP estimates based on a running particle filter, viz. particle based filter MAP (Pf-MAP) (Driessen and Boers (2008*a*)) and End Point Viterbi-Godsill MAP (EP-VGM) (Godsill *et al.* (2001)) and subsequently we compare them. We propose some efficiency improvement schemes for particle based filter MAP and study some of them further. We are then in a position to develop the smoothed marginal MAP estimator as mentioned above, when either a forward-backward or two filter smoother is used to generate the particle clouds. The smoothed marginal MAP estimator is then applied to estimate the unknown initial state of a dynamic system. This chapter is based on (Saha *et al.* (2009*c*, 2008*b*,*a*)).

Chapter 5 There are many approaches to the parameter estimation problem for state space models. In this chapter, we specifically develop algorithms for estimating the parameters without introducing any artificial dynamics. The artificial noise turns the fixed parameter into a slowly varying one. Here we introduce some new particle filtering/smoothing based schemes, where we avoid any effect of the

artificial dynamics on the (final) estimate of the parameters. We start with the idea of the augmented state space without any parameter dynamics and apply this method to an application of immense practical importance in mathematical finance, where we estimate the stochastic volatility and the model parameters of the well known Heston model with jumps (Bates (1996)). Next, we explain the idea of smoothed marginal MAP as developed in chapter 4 to estimate the parameters. Subsequently, we discuss the parameter estimation method using a marginalized particle filter. Finally, we introduce another parameter estimation method which is very effective when the available observation data is short. We also outline the possible limitation of each method. This chapter is based on (Saha *et al.* (2008*b*,*a*, 2009*a*); Aihara *et al.* (2008)).

Chapter 6 This chapter presents conclusions and recommendations on the possible directions for future research.

Chapter 2

A brief overview of particle filtering methods

2.1 Introduction

Optimal estimation problems for general state space models do not typically admit a closed form solution. However, modern Monte Carlo methods have opened the door to solve such complex estimation problems. Particle filters are a popular class of such Monte Carlo based algorithms, which solve these estimation problems numerically in a recursive manner. This chapter intends to serve as an introduction to the basic particle filter, which plays a pivotal role in our subsequent works.

2.2 Dynamic modeling and Bayesian filtering

For simplicity, consider the following nonlinear dynamic system given by

$$x_k = f(x_{k-1}, w_k) \tag{2.2.1}$$
$$y_k = h(x_k, v_k) \tag{2.2.2}$$

where x_k are the unobservable system values (the state) with initial (prior) density $p(x_0)$ and y_k are the observed values (the measurements). The process noises w_k, $k = 1, 2, \cdots$ are assumed to be independent. So are the measurement noises v_k, $k = 1, 2, \cdots$. Furthermore, (w_k) is assumed to be independent of (v_k). In this model, we assume that the probability density functions for w_k and v_k are known. The above model can also be

characterized in terms of its probabilistic description via the state transition density $p(x_k|x_{k-1})$ and the observation density $p(y_k|x_k)$. Note that, here we assume x_k is Markovian, i.e. the conditional density of x_k given the past state $x_{0:k-1} \equiv (x_0, x_1, \ldots, x_{k-1})$, depends only on x_{k-1}. Furthermore, the conditional density of y_k given the state $x_{0:k}$ and the past observations $y_{0:k-1}$, depends only on x_k. Such nonlinear dynamic systems can be found abundantly in many areas of science and engineering, such as target tracking, computer vision, terrain navigation and finance among others. The main statistical problem related to this type of state-space model is to estimate the state of the dynamic system x_k in some optimal manner from all the noisy observations $y_{1:k}$, up to time k. This is known as the filtering problem. The complete solution to this estimation problem can be given by the conditional density or filtered density $p(x_k|y_{1:k})$, containing all available statistical information. For a point estimate, one can, for example, consider the corresponding conditional mean or maximum *a posteriori* (MAP) of this filtered density. A simple application of the Bayes rule leads to

$$p\left(x_k|y_{1:k}\right) = \frac{p\left(x_k|y_{1:k-1}\right) p\left(y_k|x_k\right)}{\int p\left(x_k|y_{1:k-1}\right) p\left(y_k|x_k\right) dx_k} \tag{2.2.3}$$

$$\text{where} \quad p\left(x_k|y_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_{k-1}|y_{1:k-1}\right) dx_{k-1} \tag{2.2.4}$$

Equation (2.2.4) is generally referred to as the prediction equation and (2.2.3) as the update equation. Thus, starting from the initial density $p(x_0)$ one can, at least in principle, recursively arrive at the desired density $p(x_k|y_{1:k})$. However, analytical solution is available only for a restrictive set of cases where the posterior density can be characterized by a sufficient statistic of fixed and finite dimension. This happens, for example, when both the system and observation equation (2.2.1)–(2.2.2) are linear, driven by Gaussian white noises, in which the posterior is Gaussian and completely characterized by the conditional mean and conditional covariance. For this case, the recursive propagation of the sufficient statistic can be obtained analytically using the Kalman filter. However, due to ubiquitous presence of nonlinearity and non-Gaussianity in most practical problems, it is not possible to obtain an analytical solution. As a result, analytical approximations such as the extended Kalman filter (EKF) and Gaussian sum filter (GSF) are developed (Anderson and Moore (1979); Bagchi (1993); Jazwinski (1970)). The extended Kalman filter is based on local linearization of the state and measurement equations along the tra-

jectories. Although EKF, and its variants, have been successfully applied to many nonlinear filtering problems, whenever there is substantial nonlinearity in the system, or the noises are significantly non Gaussian, EKF exhibits very poor performances. In GSF, the posterior is approximated by a weighted sum of Gaussian densities. However, the recursive implementation of the algorithm is not trivial and the number of components in general, can grow exponentially with time (Ristic *et al.* (2004)). With easy availability of computers, other approximate methods are proposed such as using numerical integration to arrive at the solutions (Kitagawa (1987)), the unscented Kalman filter (Julier and Uhlmann (1997); Wan and van der Merwe (2000)) and the Gaussian quadrature Kalman filter (Ito and Xiong (2000)).

The Particle filtering (PF) method, on the other hand, uses the Monte Carlo simulation technique to reach a solution. Though the methods were introduced in the 1960's and 70's (Handschin and Mayne (1969); Handschin (1970); Akashi and Kumamoto (1975)), severe computational limitations may have stopped researchers to pursue that line. Another not so obvious reason is that all the early implementations were based on plain sequential importance sampling, which, as we will be describing later, suffers from 'degeneracy' over time. This phenomenon was not clearly identified and addressed until the seminal work of Gordon *et al.* (1993). Starting with this pioneering work, together with the advent of more and more powerful computers, the PF methods have started receiving enormous attention (West (1993); Liu and Chen (1998); Pitt and Shephard (1999); Doucet *et al.* (2001); Arulampalam *et al.* (2002)).

The biggest advantage of the PF is that the method can easily adapt to the nonlinearity in the model and/or non-Gaussian noises and as such, ad-hoc approximation on the model for the purpose of analytical tractability is not required. The PF methods are often referred to as sequential Monte Carlo methods (SMC's), although in strict sense PF is a sub class of SMC methods (Doucet and Johansen (2009)). Unless stated otherwise, we will use SMC and PF interchangeably in this chapter. In this method, probability distributions are represented by a cloud of particles (Monte Carlo samples). Particles are recursively generated via Monte Carlo simulation from a so called importance function, $\pi(\cdot)$, also often referred to as proposal distribution. Furthermore, each particle receives an importance weight attached to it. The resulting distributions (represented by the particle clouds) do converge to the true filtered distribution as the

Monte Carlo sample size tends to infinity (Crisan and Doucet (2002)). PF is proving to be a dependable method for stochastic dynamic estimation in real time, whose applications include target tracking, computer visions, robotics and mathematical finance among others. As a result of this popularity, many tutorials on particle filters have already been published (Doucet *et al.* (2001); Arulampalam *et al.* (2002); Chen (2003); Cappé *et al.* (2007); Doucet and Johansen (2009); Gustafsson (2009)).

This chapter serves as an initial exposition to the subject of particle filtering. As mentioned earlier, a number of tutorials/review papers are already available in the literature and as such we will not be repeating those in great detail. The organization of this chapter is as follows. We start with a very basic review of Monte Carlo methods and Importance Sampling (IS). Subsequently, we present Sequential Importance Sampling (SIS) method. We then discuss the limitations of this method and show how the Resampling step, first introduced by Gordon *et al.* (1993) in this context, can partially mitigate the problem. Next we describe a generic particle filter algorithm followed by concluding remarks.
Before proceeding to the next section, we note that the joint posterior $p(x_{0:k}|y_{1:k})$ can be recursively obtained as

$$\begin{aligned} p(x_{0:k}|y_{1:k}) &= \frac{p(y_k|x_{0:k}, y_{1:k-1})p(x_{0:k}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &= \frac{p(y_k|x_{0:k}, y_{1:k-1})p(x_k|x_{0:k-1}, y_{1:k-1})p(x_{0:k-1}|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \end{aligned} \quad (2.2.5)$$

Now, using the Markovian nature of the model given by (2.2.1)–(2.2.2), one can write equation (2.2.5) as

$$\begin{aligned} p(x_{0:k}|y_{1:k}) &= \frac{p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &\propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|y_{1:k-1}). \end{aligned} \quad (2.2.6)$$

The above recursion forms the basis of most of the particle filters.

2.3 Preliminaries

2.3.1 Basics of Monte Carlo Methods

Suppose we are able to sample N independent random variables, $x_{0:k}^{(i)} \sim p(x_{0:k}|y_{1:k})$ for $i = 1, \ldots, N$. Then the Monte Carlo method approximates

$p(x_{0:k}|y_{1:k})$ by the empirical measure (Doucet *et al.* (2001))

$$P_N(dx_{0:k}|y_{1:k}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_{0:k}^{(i)}}(dx_{0:k}), \tag{2.3.1}$$

where $\delta_{x_{0:k}^{(i)}}(dx_{0:k})$ denotes the Dirac-delta mass located in $x_{0:k}^{(i)}$. Subsequently one can approximate the marginal $p(x_k|y_{1:k})$ as

$$P_N(dx_k|y_{1:k}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_k^{(i)}}(dx_k), \tag{2.3.2}$$

and expectations of the form

$$I(g_k) = \int g_k\left(x_{0:k}\right) p\left(x_{0:k}|y_{1:k}\right) dx_{0:k} \tag{2.3.3}$$

as

$$\begin{aligned} I_N(g_k) &= \int g_k\left(x_{0:k}\right) P_N(dx_{0:k}|y_{1:k}) \\ &= \frac{1}{N}\sum_{i=1}^{N} g_k(x_{0:k}^{(i)}). \end{aligned} \tag{2.3.4}$$

This estimate is unbiased and according to the law of large numbers, I_N will almost surely converge to I (Ristic *et al.* (2004)). Moreover, if the variance of $g_k(x_{0:k})$,

$$\sigma^2 = \int (g_k(x_{0:k}) - I)^2 p\left(x_{0:k}|y_{1:k}\right) dx_{0:k}$$

is finite, the central limit theorem will hold and the estimation error will converge in distribution as

$$\lim_{N\longrightarrow\infty}\sqrt{N}(I_N - I) \sim \mathcal{N}(0, \sigma^2).$$

The error of the estimate, $e = (I_N - I)$, is of order $O(N^{-\frac{1}{2}})$, thus the rate of convergence of this estimate is independent of the dimension of the integrand. In contrast, any deterministic integration method has a rate of convergence that decreases as the dimension of the integrand increases (Doucet *et al.* (2001)). This is one of the main advantages of Monte Carlo integration methods, although the rate of convergence is very slow.

Unfortunately, it is usually not possible to sample effectively from the posterior $p(x_{0:k}|y_{1:k})$, which is typically multivariate, nonstandard, and only known up to a proportionality constant (Doucet *et al.* (2001)). A possible solution addressed in the statistical literature, is to use importance sampling methods. The idea behind this is explained next.

2.3.2 Importance Sampling (IS)

Suppose we can not generate samples directly from $p(x_{0:k}|y_{1:k})$ to estimate I_N using equation (2.3.4). One can introduce an arbitrary distribution $\pi(x_{0:k}|y_{1:k})$, from which it is easy to sample and whose support covers the support of $p(x_{0:k}|y_{1:k})$. This distribution is known as importance distribution (also often referred to as proposal distribution) and the integration in equation (2.3.3) can be rewritten as

$$I(g_k) = \int g_k\left(x_{0:k}\right) \bar{w}(x_{0:k}) \pi\left(x_{0:k}|y_{1:k}\right) dx_{0:k}, \tag{2.3.5}$$

where $\bar{w}(x_{0:k})$ is known as the importance weight, which is given by

$$\bar{w}(x_{0:k}) = \frac{p(x_{0:k}|y_{1:k})}{\pi(x_{0:k}|y_{1:k})}, \tag{2.3.6}$$

and assumed to be upper bounded. However, often the target distribution $p(x_{0:k}|y_{1:k})$ is known only upto a normalizing factor, particularly in Bayesian statistical inference problems (Cappé *et al.* (2007)). Then the importance weight $\bar{w}(x_{0:k})$ is known only upto a scaling factor and $I(g_k)$ in equation (2.3.5) would require the knowledge of the actual normalizing factor. This requirement can be avoided as follows. Suppose $p(x_{0:k}|y_{1:k}) \propto q(x_{0:k}|y_{1:k})$. Defining new importance weight $w(x_{0:k})$ as

$$w(x_{0:k}) = \frac{q(x_{0:k}|y_{1:k})}{\pi(x_{0:k}|y_{1:k})}, \tag{2.3.7}$$

$I(g_k)$ can now be written as

$$I(g_k) = \frac{\int g_k\left(x_{0:k}\right) w(x_{0:k}) \pi\left(x_{0:k}|y_{1:k}\right) dx_{0:k}}{\int w(x_{0:k}) \pi\left(x_{0:k}|y_{1:k}\right) dx_{0:k}}. \tag{2.3.8}$$

Accordingly, one can simulate N independent samples (also known as particles), $x_{0:k}^{(i)} \sim \pi(x_{0:k}|y_{1:k})$, for $i = 1, \ldots, N$ and the Monte Carlo approxi-

mation of the integration in equation (2.3.8) can now be given by

$$\begin{aligned} \widehat{I}_N(g_k) &= \frac{\frac{1}{N}\sum_{i=1}^{N} g_k(x_{0:k}^{(i)})w(x_{0:k}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} w(x_{0:k}^{(i)})} \\ &= \sum_{i=1}^{N} g_k(x_{0:k}^{(i)})\widetilde{w}_k^{(i)}, \end{aligned} \tag{2.3.9}$$

where $\widetilde{w}_k^{(i)}$ are the normalized importance weights, given by

$$\widetilde{w}_k^{(i)} = \frac{w(x_{0:k}^{(i)})}{\sum_{i=1}^{N} w(x_{0:k}^{(i)})}. \tag{2.3.10}$$

Asymptotically $\widehat{I}_N$ converges to I almost surely by the strong law of large numbers under the following weak assumptions (Geweke (1989)):
Let $x_{0:k} \in \mathbb{X}^k \subseteq \mathbb{R}^k$. Then

Assumption (1) : The product of the prior density, $p(x_{0:k}|y_{1:k-1})$, and the likelihood function, $p(y_k|x_{0:k}, y_{1:k-1})$, is proportional to a proper probability density function defined on $\mathbb{X}^k$.

Assumption (2) : $\{x_{0:k}^{(i)}\}_{i=1}^{\infty}$ is a sequence of i.i.d. random vectors, the common distribution having a probability distribution function $\pi(x_{0:k}|y_{1:k})$.

Assumption (3) : The support of $\pi(x_{0:k}|y_{1:k})$ includes $\mathbb{X}^k$.

Assumption (4) : $I(g_k)$ exists and is finite.

A central limit theorem with a convergence rate still independent of the dimension of the integrand can also be obtained under the following additional assumptions:

Assumption (5) : $\mathbb{E}[\widetilde{w}_k] < \infty$.

Assumption (6) : $\mathbb{E}[\widetilde{w}_k\ g_k^2(x_{0:k})] < \infty$.

The integration $\widehat{I}_N(g_k)$ in equation (2.3.9) can be construed to be an integration of the function $g_k(x_{0:k})$ with respect to the empirical measure $\widehat{P}_N(dx_{0:k}|y_{1:k})$ as

$$\widehat{I}_N(g_k) \quad = \quad \int g_k\left(x_{0:k}\right) \widehat{P}_N(dx_{0:k}|y_{1:k}), \tag{2.3.11}$$

where

$$\widehat{P}_N(dx_{0:k}|y_{1:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \delta_{x_{0:k}^{(i)}}(dx_{0:k}), \tag{2.3.12}$$

is the weighted particle approximation of the posterior $p(x_{0:k}|y_{1:k})$.

The importance sampling method as described above, is often used in general Monte Carlo integration methods. However, in its simplest form, this is not amenable to recursive estimation, as with the arrival of new measurement, one has to recompute the importance weights over the entire state sequence. As a result, the computational complexity increases with time. A strategy to address this problem in the context of sequential estimation is described next.

2.3.3 Sequential Importance Sampling (SIS)

Suppose at time $k-1$, we have a weighted particle approximation of the posterior $p(x_{0:k-1}|y_{1:k-1})$ as $\widehat{P}_N(dx_{0:k-1}|y_{1:k-1}) = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \delta_{x_{0:k-1}^{(i)}}(dx_{0:k-1})$. With the arrival of a new measurement y_k, we wish to approximate $p(x_{0:k}|y_{1:k})$ with a new set of samples. If we choose the importance function $\pi(x_{0:k}|y_{1:k})$ such that it admits the importance function $\pi(x_{0:k-1}|y_{1:k-1})$ as its marginal distribution, i.e.

$$\pi(x_{0:k}|y_{1:k}) = \pi(x_{0:k-1}|y_{1:k-1})\pi(x_k|x_{0:k-1}, y_{1:k}) \tag{2.3.13}$$

then one can obtain the particles (samples) $x_{0:k}^{(i)} \sim \pi(x_{0:k}|y_{1:k})$ by augmenting each of the existing particles $x_{0:k-1}^{(i)} \sim \pi(x_{0:k-1}|y_{1:k-1})$ with a new state $x_k^{(i)}$ drawn according to $\pi(x_k|x_{0:k-1}, y_{1:k-1})$. Moreover, using equation (2.2.6) together with this importance function given by equation (2.3.13), one can evaluate the (unnormalized) importance weights recursively as

$$w_k^{(i)} \propto \widetilde{w}_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{\pi(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})}. \tag{2.3.14}$$

Ideally the importance function should be the posterior itself (i.e $p(x_{0:k}|y_{1:k})$), but this is the quantity of our interest, which is unknown. It is known that using importance function of the form as in equation (2.3.13), the variance of the importance weights can only increase over time (Kong *et al.* (1994)) and this leads to a so called degeneracy problem, where distribution of the importance weights become more and more skewed. In practical terms, it means that after a few iterations, most of the particles will carry very insignificant weights, whereas most weights will be carried by a few particles. Consequently, the algorithm fails to represent the posterior distributions adequately and effectively a significant computational effort is wasted for updating those particles with very low weights, as their contributions to the approximation of the posterior is almost negligible (Doucet *et al.* (2001); Ristic *et al.* (2004); Doucet *et al.* (2000); Cappé *et al.* (2007)). A suitable measure of this degeneracy is usually given by the effective sample size (ESS) N_{eff} introduced in Kong *et al.* (1994) and defined as follows:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\widetilde{w}_k^{(i)})^2}, \tag{2.3.15}$$

with $1 \leq N_{eff} \leq N$. A small N_{eff} indicates a severe degeneracy (Ristic *et al.* (2004)). This degeneracy phenomenon can (partially) be mitigated by the introduction of a new step called resampling, which we will describe next.

2.3.4 Resampling

Resampling step is a key and essential element for successful implementation of the PF. The intuitive idea behind resampling is to statistically eliminate samples with low importance weights and replicate samples with higher importance weights. In formal term, resampling step replaces the weighted empirical measure $\widehat{P}_N(dx_{0:k}|y_{1:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \delta_{x_{0:k}^{(i)}}(dx_{0:k})$ by the unweighted measure

$$\widetilde{P}_N(dx_{0:k}|y_{1:k}) = \frac{1}{N}\sum_{i=1}^{N} N_k^{(i)} \delta_{x_{0:k}^{(i)}}(dx_{0:k}), \tag{2.3.16}$$

where $N_k^{(i)}$ is the number of offspring associated to particle $x_{0:k}^{(i)}$ (Doucet *et al.* (2001)); $N_k^{(i)}$ is an integer number such that $\sum_{i=1}^{N} N_k^{(i)} = N$. If $N_k^{(j)} = 0$ then the particle $x_{0:k}^{(j)}$ dies. The surviving particles $x_{0:k}^{(i)}$, i.e. those

particles, for which $N_k^{(i)} > 0$, are approximately distributed according to $P(x_{0:k}|y_{1:k})$. Different unbiased resampling schemes (e.g. Systematic Resampling, Residual Resampling and Multinomial Resampling) have been proposed in the literature. For a review on their comparative performances, please refer to the article by Hol *et al.* (2006).

Although one can reduce the effect of degeneracy through the resampling step, the latter introduces other problems. The particles with high importance weights are statistically selected many times, leading to a loss of diversity among the particles. This is known as 'sample impoverishment' and it can be severe when the process noise is very low (Ristic *et al.* (2004)). To address this problem, schemes like MCMC move step (Carlin *et al.* (1992)), resample-move algorithm (Gilks and Berzuini (2001)) and regularization step (Musso *et al.* (2001)) have been proposed in the literature. However, we will not discuss them here any further. The other problem with resampling is that as the particles interact via this resampling step, parallel implementation of SIS (with resampling) becomes difficult. Resampling step also introduces some additional variance on the estimate, but it can be seen to provide stability later (Doucet *et al.* (2001)). Consequently, it is more sensible in practice to resample only when the effective sample size (defined in equation (2.3.15)) as a manifestation of degeneracy falls below a pre specified threshold N_{Thr}. From a theoretical point of view, after resampling step, the simulated trajectories are no longer statistically independent (Doucet *et al.* (2000)) and consequently, the convergence results are much more involved than that of SIS. For more details about the convergence issues, one may consult Crisan and Doucet (2002), Hu *et al.* (2008) and Del Moral (2004).

2.4 A Generic SMC Algorithm

Having discussed about all the necessary ingredients, we are now in a position to describe a generic particle filter algorithm. This is explained below:

Recursively over time $k = 0, 1, 2, \ldots$

`For` $i = 1, \ldots, N$, `where` N `is the total number of particles,`

- `sample` $x_k^{(i)} \sim \pi\left(x_k|x_{0:k-1}^{(i)}, y_{1:k}\right)$ `and set` $x_{0:k}^{(i)} \triangleq \left(x_{0:k-1}^{(i)}, x_k^{(i)}\right)$
- `evaluate the corresponding importance weights` $w_k^{(i)}$ `according to (2.3.14)`

- `normalized the importance weights` $\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}}$

To avoid carrying the trajectories with small normalized importance weights and to concentrate upon the ones with large weights, the effective sample size N_{eff} is used to decide resampling.

- `If` N_{eff} `is below a specified threshold` N_{Thr}`, resample from` $\{x_k^{(i)}\}_{i=1}^{N}$ `with probabilities` $\{\tilde{w}_k^{(i)}\}_{i=1}^{N}$ `keeping the sample size still to be` N `and assign equal weights` $1/N$`.`

2.5 Concluding Remarks

Estimating the optimal state of a nonlinear dynamic model sequentially over time is of paramount importance in different science and engineering applications. However, except for a few special cases, the closed form solution is not in general available and can only be approximated. Particle filtering methods have become very popular as a powerful tool to accomplish this goal numerically. In these methods, the posterior distribution is approximated by a cloud of weighted random particles and as new observation arrives, the cloud propagates over time. In this chapter, we have provided a basic exposition to the particle filtering algorithm. We have described a generic particle filter, typically consisting of three basic building blocks - (1) generating the samples sequentially, (2) updating the weights of the corresponding samples using SIS principle and (3) resampling whenever necessary to prevent degeneracy. These ingredients are vary modular in nature and they can be easily implemented.

After describing the working principle of a generic particle filter here, we are now ready to move on to the next chapter which focuses on the issues with sequential importance sampling.

Chapter 3

Gaussian proposal density using moment matching in particle filtering methods

3.1 Introduction

Consider the dynamic system as given by equations (2.2.1)-(2.2.2). The typical statistical problem related to this type of state-space model is to estimate the unobserved system value x_k in some optimal manner from all the observations $y_{1:k} \equiv (y_1, y_2, \ldots, y_k)$, up to time k, or equivalently, estimate the conditional density (also known as filtered density) $p(x_k|y_{1:k})$. As discussed in the previous chapters, the closed form solution can not be obtained in general. However, this posterior can be elegantly approximated using particle filtering methods.

Particle Filter (PF), which is a class of simulation based sequential Monte Carlo (SMC) methods, provides the target filter density $p(x_k|y_{1:k})$ in the form of a cloud of particles (Handschin and Mayne (1969); Akashi and Kumamoto (1975); Gordon *et al.* (1993); West (1993); Pitt and Shephard (1999); Doucet *et al.* (2001); Arulampalam *et al.* (2002)). The biggest advantage of the PF is that it can easily adapt to a nonlinearity in the model and/or non-Gaussian noises. The efficiency of the PF algorithm depends on the so-called importance function $\pi(\cdot)$, also often referred to as the proposal distribution, used to generate the particles. Naturally, the appropriate selection of this importance function lies at the heart of PF's implementation and forms the main issue addressed in this chapter. It has

been shown by Doucet *et al.* (2000) that the importance function of the form $\pi = p(x_k|x_{k-1}, y_k)$, is optimal in a certain sense. In practice, however, there are two major prohibitive drawbacks for using this importance function. Firstly, drawing samples according to $p(x_k|x_{k-1}, y_k)$ is, in general, difficult. Secondly, it is also difficult to get an analytical expression which is needed for the weight update.

In this chapter, we propose a Gaussian approximation of $p(x_k|x_{k-1}, y_k)$ as the importance function by matching the exact moments up to second order of the distribution of (x_k, y_k) conditional on x_{k-1} (Saha *et al.* (2009*b*, 2007, 2006)). Recently, Doucet *et al.* (2000) and Guo *et al.* (2005) also proposed similar importance functions. However, Doucet *et al.* (2000) use the linearized approximation of the observation model to calculate the moments, while Guo *et al.* (2005) approximate the moments by different numerical methods such as the Gaussian-Hermite quadrature rule or the Julier-Uhlmann quadrature rule. Besides the possibility that the use of exact moments may lead to better approximation of the optimal proposal distribution, our method has one distinct advantage over the one in Guo *et al.* (2005), namely that, it is computationally less demanding. We also note that, the dynamical systems considered by Doucet *et al.* (2000) and Guo *et al.* (2005) are with additive Gaussian noise processes, whereas our method would work for more general models, for example, when the Gaussian noise in the state equation (2.2.1) is not necessarily additive but the observation model (2.2.2) is a polynomial. For comparison we have also considered unscented particle filter (van der Merwe *et al.* (2000)) which, too, works for general dynamical systems. Our experimental results with additive Gaussian noise processes show that the overall performance of our proposal function is better than that of the other proposals, considering the trade off between the RMSE and the computational load.

The rest of the chapter is organized as follows. In section 3.2 the importance function is discussed briefly. We describe our proposed importance function using the exact moments in section 3.3. Construction of other importance functions as proposed by Doucet *et al.* (2000), Guo *et al.* (2005) and van der Merwe *et al.* (2000) are briefly reviewed in section 3.4. Implementation issues of our proposed method are discussed in section 3.5. Section 3.6 contains the numerical comparison results of these methods based on three examples. The first two are with additive Gaussian noise processes – one with a polynomial (section 3.6.1) and the other with a non-polynomial (section 3.6.2) observation equation while the third one is with

non Gaussian noise process (section 3.6.3). In section 3.7, we compare the different proposal densities in terms of Kullback-Leibler divergence. Finally, section 3.8 concludes the chapter.

3.2 The importance function in PF

Usually in practice, the importance function is taken to be the transition density, i.e., $\pi\left(x_k \left| x_{0:k-1}^{(i)}, y_{1:k}\right.\right) = p\left(x_k \left| x_{k-1}^{(i)}\right.\right)$, because it is easily available from the model (Arulampalam *et al.* (2002)). However, it is well known that particle filter algorithm using this importance function suffers from the degeneracy problem, that is to say, the variance of the importance weights can only increase over time. Intuitively, if the measurement is very informative, a lot of samples are wasted by exploring regions of low importance. To make the method more effective, importance functions of the form $\pi = p(x_k|x_{k-1}, y_k)$, i.e., the one which incorporates both the system and observation processes are suggested in (Doucet *et al.* (2000); Liu and Chen (1998)). Additionally, it has been shown by Doucet *et al.* (2000) that the importance function $p\left(x_k \left| x_{k-1}^{(i)}, y_k\right.\right)$ addresses the degeneracy issue by minimizing the variance of the (unnormalized) importance weight $w_k^{(i)}$ conditional upon $x_{0:k-1}^{(i)}$ and $y_{1:k}$. It is, however, very difficult to get this optimal importance function $p\left(x_k \left| x_{k-1}, y_k\right.\right)$, barring a few special cases. One such case is the celebrated Heston model with jumps (also known as Bates model (Bates (1996))) for (unobserved) volatility of a stock price process. The detailed calculation can be found in Appendix B.

In general, though, as mentioned before, this choice is not practical because it is neither easy to generate samples from this distribution nor to get the analytical expression (needed for the weight update equation). In the following section, we propose an approximation of this as the importance function.

3.3 Importance function based on exact moment matching (EMM)

Suppose the system dynamics are given by (2.2.1)–(2.2.2). We further assume the following.

Assumption (A) : All the moments of (x_k, y_k) conditional on x_{k-1} up to second order, i.e., $E(x_k|x_{k-1})$, $E\left(x_k\, x_k^T|x_{k-1}\right)$, $E(y_k|x_{k-1})$, $E\left(y_k\, y_k^T|x_{k-1}\right)$, and $E\left(x_k\, y_k^T|x_{k-1}\right)$ are known.

To determine the importance function, to be used in conjunction with the particle filtering algorithm, we proceed as follows. We approximate the joint distribution of (x_k, y_k), conditional on x_{k-1}, by a Gaussian distribution with matching moments up to the second order. Let the corresponding mean $\mu^{(k)}$ and the covariance $\Sigma^{(k)}$ be given by

$$\mu^{(k)} = \begin{pmatrix} \mu_1^{(k)} \\ \mu_2^{(k)} \end{pmatrix} \quad \text{and} \quad \Sigma^{(k)} = \begin{pmatrix} \Sigma_{11}^{(k)} & \Sigma_{12}^{(k)} \\ \left(\Sigma_{12}^{(k)}\right)^T & \Sigma_{22}^{(k)} \end{pmatrix}. \tag{3.3.1}$$

Note that $\mu^{(k)}$ and $\Sigma^{(k)}$ can be calculated from the moments which are assumed to be known from Assumption (A). Subsequently, we take the importance function to be the conditional distribution of x_k, given (x_{k-1}, y_k), derived from the approximated Gaussian distribution above. In other words, we take $\pi(x_k|x_{0:k-1}^{(i)}, y_{0:k}) \sim \mathcal{N}(m_k, \Sigma_k)$, where

$$m_k = \mu_1^{(k)} + \Sigma_{12}^{(k)} \left[\Sigma_{22}^{(k)}\right]^{-1} (y_k - \mu_2^{(k)}) \tag{3.3.2}$$

$$\Sigma_k = \Sigma_{11}^{(k)} - \Sigma_{12}^{(k)} \left[\Sigma_{22}^{(k)}\right]^{-1} \left(\Sigma_{12}^{(k)}\right)^T. \tag{3.3.3}$$

We note here that a sufficient condition for Assumption (A) to hold is :

Condition 1 : Both $E(y_k|x_k)$ and $E\left(y_k\, y_k^T|x_k\right)$ are polynomials in x_k of degree at most m and all conditional moments of x_k, given x_{k-1}, are known up to order m, where m is an arbitrary positive integer.

In the special case when the noise processes in (2.2.1)–(2.2.2) are additive Gaussian, suppose the system dynamics are given by

$$x_k = f(x_{k-1}) + w_k, \quad w_k \sim \mathcal{N}(0, Q) \tag{3.3.4}$$

$$y_k = h(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, R), \quad k = 1, 2, \ldots \tag{3.3.5}$$

then the conditional moments of all order of x_k given x_{k-1} are known. If, in addition, $h(\cdot)$ in (3.3.5) is a polynomial, then Condition 1 would be satisfied and hence Assumption (A) would hold. The precise formulas for the quantities can be found in section 3.5.

3.4 Other Gaussian importance functions

There are other Gaussian importance functions proposed in the literature. We mention three of them here. The first two are based on the Gaussian approximation of the optimal importance function $p\left(x_k|x_{k-1}, y_k\right)$ which are similar in idea to EMM, but the moments are approximated in different ways. The third one, on the other hand, uses a bank of unscented Kalman filters to obtain the proposal density.

3.4.1 Importance function by linearization (LIN)

In an earlier paper, Doucet *et al.* (2000) consider a dynamical system with additive Gaussian noise, given by (3.3.4)–(3.3.5). Observing that the optimal importance function $p\left(x_k|x_{k-1}, y_k\right)$ is Gaussian when $h(\cdot)$ in the observation model (3.3.5) is linear, the authors linearize the observation equation (3.3.5) to obtain

$$y_k \approx h(f(x_{k-1})) + C_k(x_k - f(x_{k-1})) + v_k \tag{3.4.1}$$

where $C_k = \frac{\partial h}{\partial x_k}(f(x_{k-1}))$. Subsequently, they use the corresponding Gaussian distribution as the importance function, i.e., $\pi\left(x_k|x_{0:k-1}^{(i)}, y_{1:k}\right) \sim \mathcal{N}(m_k, V_k)$, where

$$\begin{aligned} V_k^{-1} &= Q^{-1} + C_k^T R^{-1} C_k & (3.4.2)\\ m_k &= V_k Q^{-1} f(x_{k-1}) + \\ &\quad V_k C_k^T R^{-1}\left(y_k - h(f(x_{k-1})) + C_k f(x_{k-1})\right). & (3.4.3) \end{aligned}$$

This essentially reduces to approximating the conditional distribution of (x_k, y_k), given x_{k-1}, by the Gaussian distribution with mean vector μ^* and covariance matrix Σ^* given by

$$\mu^* = \begin{pmatrix} f(x_{k-1}) \\ h(f(x_{k-1})) \end{pmatrix} \text{ and } \Sigma^* = \begin{pmatrix} Q & QC_k^T \\ C_kQ & C_kQC_k^T + R \end{pmatrix}. \tag{3.4.4}$$

See Appendix A for details.

3.4.2 Numerically approximated moment matching

In a more recent article, Guo *et al.* (2005) consider the same dynamical system given by (3.3.4)–(3.3.5). The importance function proposed

by them is also in effect derived from a Gaussian approximation of the joint distribution of (x_k, y_k) conditional on x_{k-1}. Guo *et al.* (2005), however, approximate the moments in (3.3.1) by various numerical techniques, such as the Gauss-Hermite quadrature (GHQ) rule and the Julier-Uhlmann quadrature (JUQ) rule. For example, according to GHQ

$$\int_{-\infty}^{\infty} g(x) \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-|x|^2} dx = \sum_{i=1}^{m} \omega_i g(x_i),$$

where the number (m) and the location (x_i's) of the abscissas and corresponding optimal weights (ω_i's) can be chosen beforehand (Golub (1973)). For instance, when $m = 3$ the x_i's and ω_i's are given by

x_i	0	$\pm\sqrt{3}$
ω_i	2/3	1/6

.

According to JUQ (see, e.g., Julier *et al.* (2000)) an n-dimensional standard Gaussian distribution with covariance e is approximated by a discrete distribution taking values in $\{z_1, \ldots, z_{2n+1}\}$ with corresponding probabilities $P(z_k)$ given by

$$\begin{aligned}
&z_k = (\sqrt{(n+\kappa)e})_k && P(z_k) = \frac{1}{2(n+\kappa)} && \text{if } 1 \leq k \leq n,\\
&z_k = -z_{k-n} && P(z_k) = \frac{1}{2(n+\kappa)} && \text{if } n+1 \leq k \leq 2n,\\
&z_k = 0 && P(z_k) = \frac{2\kappa}{2(n+\kappa)} && \text{if } k = 2n+1
\end{aligned}$$

where κ is the scaling parameter and $(\sqrt{(n+\kappa)e})_k$ is the kth row or column of the matrix square root of $(n+\kappa)e$. Subsequently,

$$\int g(x) \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{|x|^2}{2}} dx = \sum_{k=1}^{2n+1} g(z_k) P(z_k).$$

We refer the reader to the original article for the details.

3.4.3 Unscented particle filter (UPF)

The unscented particle filter algorithm of van der Merwe *et al.* (2000) is suitable for general dynamical systems given by (2.2.1)–(2.2.2). The

main idea here is that the proposal density is taken as the output of a separately running unscented Kalman filter (UKF). In this method, with new observation y_k, a separate unscented Kalman filter is propagated for each particle, to generate the Gaussian proposal density as

$$\pi(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k}) = p_{UPF}(x_k^{(i)}|x_{0:k-1}^{(i)} y_{1:k}) = \mathcal{N}(\bar{x}_k^{(i)}, \bar{P}_k^{(i)}).$$

For exact expressions of $\bar{x}_k^{(i)}$ and $\bar{P}_k^{(i)}$, please see van der Merwe *et al.* (2000). Next, one samples the ith particle from this density. It is worthwhile to note that unlike the extended Kalman filter, which approximates the models, the UKF works directly on the nonlinear models based on the unscented transformation (UT) method. Using the UT, one can calculate the mean and variance of a nonlinearly transformed random vector as follows: Let X be a n dimensional random vector with mean $\bar{x}$ and covariance P. Suppose, we want to calculate the mean and covariance of $Y = f(X)$. The UT chooses a set of $2n+1$ deterministic weighted samples $(W_k, \chi_k)_{k=1}^{2n+1}$, also known as sigma points, such that they completely capture the true mean and covariance of X. These weighted samples are given by (van der Merwe *et al.* (2000))

$$\begin{aligned}
\chi_k &= \bar{x} + (\sqrt{(n+\kappa)P})_k \qquad & W_k &= \frac{1}{2(n+\kappa)} & &\text{if } 1 \leq k \leq n,\\
\chi_k &= \bar{x} - (\sqrt{(n+\kappa)P})_k \qquad & W_k &= \frac{1}{2(n+\kappa)} & &\text{if } n+1 \leq k \leq 2n,\\
\chi_k &= \bar{x} & W_k &= \frac{2\kappa}{2(n+\kappa)} & &\text{if } k = 2n+1
\end{aligned}$$

where κ is the scaling parameter and $(\sqrt{(n+\kappa)P})_k$ is the kth row or column of the matrix square root of $(n+\kappa)P$. W_k is the weight associated with the kth point such that $\sum_{k=1}^{2n+1} W_k = 1$. The mean and variance of Y are then approximated as

$$\begin{aligned}
\bar{y} &= \sum_{k=1}^{2n+1} W_k f(\chi_k)\\
P_y &= \sum_{k=1}^{2n+1} W_k (f(\chi_k) - \bar{y})(f(\chi_k) - \bar{y})^T.
\end{aligned}$$

These estimates are accurate to the second order of the Taylor series expansion of $f(\cdot)$. For details, please refer to Julier *et al.* (2000).

3.5 Implementation of the EMM

Clearly, the EMM as described in section 3.3 can be implemented if the Assumption (A) holds. A proper classification of models for which Assumption (A) holds is not very easy. However, as mentioned in section 3.3, if the dynamical system is given by (3.3.4)–(3.3.5) with $h(\cdot)$ in equation (3.3.5) a polynomial function, then EMM can be implemented. This is shown below:

The conditional moments of x_k given x_{k-1} can be derived as follows.

$$E\left(x_k^m|x_{k-1}\right) = \sum_{r=0}^{m} \binom{m}{r} \left[f(x_{k-1})\right]^r E(w_k^{m-r}) = \sum_{r=0}^{m} \binom{m}{r} \left[f(x_{k-1})\right]^r \mu_{m-r}, \tag{3.5.1}$$

where μ_j is the j-th (raw) moment of the $\mathcal{N}(0,Q)$ distribution, given by

$$\mu_{2k+1} = 0 \quad \text{and} \quad \mu_{2k} = \frac{(2k)!}{2^k\, k!}\, Q^k, \quad \text{for } k = 0, 1, 2, \ldots. \tag{3.5.2}$$

When $h(x)$ is a polynomial of the form, $h(x) = \sum_{r=0}^{n} a_r\, x^r$, both $E\left(y_k|x_k\right)$ and $E\left(y_k\, y_k^T|x_k\right)$ are polynomials in x_k, thus satisfying Condition 1. Subsequently, we have

$$\begin{aligned} E\left(y_k|x_{k-1}\right) &= \sum_{m=0}^{n} a_m\, E\left(x_k^m|x_{k-1}\right) \\ &= \sum_{m=0}^{n}\sum_{r=0}^{m} a_m \binom{m}{r} \left[f(x_{k-1})\right]^r \mu_{m-r} \end{aligned} \tag{3.5.3}$$

$$\begin{aligned} E\left(x_k\, y_k^T|x_{k-1}\right) &= \sum_{m=0}^{n} a_m\, E\left(x_k^{m+1}|x_{k-1}\right) \\ &= \sum_{m=0}^{n}\sum_{r=0}^{m+1} a_m \binom{m+1}{r} \left[f(x_{k-1})\right]^r \mu_{m+1-r} \end{aligned} \tag{3.5.4}$$

$$\begin{aligned} E\left(y_k\, y_k^T|x_{k-1}\right) &= \sum_{m=0}^{n}\sum_{l=0}^{n} a_m\, a_l\, E\left(x_k^{m+l}|x_{k-1}\right) \\ &= \sum_{m=0}^{n}\sum_{l=0}^{n}\sum_{r=0}^{m+l} a_m\, a_l \binom{m+l}{r} \times \\ &\qquad \times \left[f(x_{k-1})\right]^r \mu_{m+l-r}\ . \end{aligned} \tag{3.5.5}$$

Then equations (3.5.1)–(3.5.5) ensure the validity of Assumption (A). In this context, it is worthwhile to note that the observation model is not always given explicitly. Instead, it is given as a high fidelity algorithmic code such as in Finite element method (FEM) and Computational fluid dynamics (CFD). Simulations using these codes, in general, require huge computational time. To reduce the computational burden often a so called "*Surrogate model*" is used. "*Response Surface Methodology*" (RSM) is one such popular technique in which the true observation model (given by the algorithmic code) is approximated in some optimal manner by a lower order polynomial in state (Giunta *et al.* (1995); Kalnins *et al.* (2006); Koehler and Owen (1996)). In such cases, EMM can be readily applied.

When the exact values of the quantities in equation (3.3.1) cannot be calculated, we propose to approximate the observation equation by one of polynomial form and implement the EMM to derive the importance function. For instance, consider a real-valued dynamical system given by (3.3.4)–(3.3.5). We assume further that the function $h(\cdot)$ is n times differentiable. We approximate $h(\cdot)$ locally by its n-th degree Taylor polynomial around $x_k^* = f(x_{k-1})$ to get the following observation equation.

$$y_k \;=\; \sum_{m=0}^{n} a_m \,(x_k - x_k^*)^m + v_k \quad \text{with} \quad a_m \;=\; \frac{1}{m!}\left(\frac{\partial^m h\,(x_k)}{\partial x_k^m}\right)_{x_k = x_k^*}. \tag{3.5.6}$$

Then the quantities in (3.3.1) can be approximated by the corresponding quantities for the dynamical system governed by equations (3.3.4) and (3.5.6). Subsequently noting that the conditional distribution of $(x_k - x_k^*)$ given x_{k-1} is $\mathcal{N}(0, Q)$, we can calculate the following moments.

$$\begin{aligned} E\,(y_k|x_{k-1}) \;&=\; \sum_{m=0}^{n} a_m\, E\,[(x_k - x_k^*)^m|x_{k-1}] \\ &=\; \sum_{m=0}^{n} a_m\, \mu_m, \end{aligned} \tag{3.5.7}$$

$$\begin{aligned} E\left(y_k\, y_k^T|x_{k-1}\right) \;&=\; \sum_{m=0}^{n}\sum_{l=0}^{n} a_m\, a_l\, E\left[(x_k - x_k^*)^{m+l}|x_{k-1}\right] \\ &=\; \sum_{m=0}^{n}\sum_{l=0}^{n} a_m\, a_l\, \mu_{m+l}, \end{aligned} \tag{3.5.8}$$

$$
\begin{aligned}
E\left(x_k\, y_k^T | x_{k-1}\right) &= \sum_{m=0}^{n} a_m\, E\left[x_k (x_k - x_k^*)^m | x_{k-1}\right] \\
&= \sum_{m=0}^{n} a_m\, E\left[(x_k - x_k^*)^{m+1} | x_{k-1}\right] + \\
&\qquad + \; x_k^* \sum_{m=0}^{n} a_m\, E\left[(x_k - x_k^*)^m | x_{k-1}\right] \\
&= \sum_{m=0}^{n} a_m\, \mu_{m+1} + x_k^* \sum_{m=0}^{n} a_m\, \mu_m, \qquad (3.5.9)
\end{aligned}
$$

where μ_j's are given by (3.5.2). A pseudo algorithm for non-polynomial observation model can be given as follows.

Algorithm

- `Set the degree (`n`) of Taylor polynomial in (3.5.6)`
- `Set the threshold sample size,` N_{thr}

  `At time step` k
- `Compute all coefficients of polynomial using (3.5.6)`
- `Compute all the moments in (3.3.1) using (3.5.7)-(3.5.9) together with (3.5.1)-(3.5.2)`
- `Construct Gaussian proposal density` $\pi\left(x_k | x_{0:k-1}, y_{1:k}\right) \sim \mathcal{N}\left(m_k, \Sigma_k\right)$ `from (3.3.2) - (3.3.3).`
- `For` $i = 1, ..., N,$ `sample` $\widetilde{x}_k^{(i)} \sim \pi\left(x_k | x_{0:k-1}^{(i)}, y_{1:k}\right)$ `and set` $\widetilde{x}_{0:k}^{(i)} \triangleq \left(x_{0:k-1}^{(i)}, \widetilde{x}_k^{(i)}\right)$.
- `For` $i = 1, ..., N,$ `assign the importance weights upto a normalizing constant`

$$
\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p\left(y_k | \widetilde{x}_k^{(i)}\right) p\left(\widetilde{x}_k^{(i)} | \widetilde{x}_{k-1}^{(i)}\right)}{\pi\left(\widetilde{x}_k^{(i)} | \widetilde{x}_{0:k-1}^{(i)}, y_{1:k}\right)}
$$

- `For` $i = 1, ..., N$, `normalize the importance weights`

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N} \tilde{w}_k^{(i)}}$$

- `Evaluate effective sample size`

$$N_{eff} = 1/\sum_{i=1}^{N} \left(w_k^{(i)}\right)^2$$

- `If` $N_{eff} > N_{thr}$`, resample.`

Note that the approach proposed above extends the methodology used by Doucet *et al.* (2000) where the observation equation is approximated by the first degree Taylor polynomial, whereas we consider polynomials of higher degree. It is also worthwhile to note the difference between the approach followed by Guo *et al.* (2005) and the one proposed above. Guo *et al.* (2005) work with the given nonlinear model and during setting up of the Gaussian importance density, they approximate the moments. We, on the other hand, first approximate the observation equation with a n-th degree polynomial and further derive the Gaussian importance density using the exact moments (based on the approximated polynomial model). In the following section (section 3.6), we present some illustrative numerical examples.

3.6 Numerical simulation results

In this section, we first consider two examples – one with a polynomial observation model and the other with a non-polynomial model – and compare the filtered estimates obtained by different methods. In both examples we consider additive Gaussian noise processes. Finally, we consider a model with non Gaussian process noise and compare the filter estimates as obtained by EMM and UPF.

3.6.1 Polynomial observation model

As in Doucet *et al.* (2000) we consider the system dynamics to be given by (3.3.4)–(3.3.5) with

$$f(x_{k-1}) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) \tag{3.6.1}$$

$$h(x_k) = \frac{x_k^2}{20}. \tag{3.6.2}$$

In our simulations, we set $Q = 10$ and $R = 1$ and starting with $x_0 \sim \mathcal{N}(0, 5)$, generate a time series data of length 100. Given only the noisy observations y_k, the particle filter algorithm is performed with the importance functions described in section 3.4 (LIN, GHQ, JUQ, UPF) and the new one (EMM) proposed in section 3.3. We estimate the state sequence $x_k, k = 1, 2, \ldots, 100$, with all the different methods mentioned above.

Note that, in this case, the differences in the moments used in EMM and LIN can be clearly seen. The moments used in EMM, as given in (3.3.1), are

$$\mu^{(k)} = \begin{pmatrix} f(x_{k-1}) \\ \frac{f^2(x_{k-1})}{20} + \frac{Q}{20} \end{pmatrix} \quad \text{and} \quad \Sigma^{(k)} = \begin{pmatrix} Q & \frac{f(x_{k-1})Q}{10} \\ \frac{f(x_{k-1})Q}{10} & \frac{f^2(x_{k-1})Q}{100} + \frac{Q^2}{200} + R \end{pmatrix},$$

while the moments used in LIN, as given in (3.4.4), are

$$\mu^* = \begin{pmatrix} f(x_{k-1}) \\ \frac{f^2(x_{k-1})}{20} \end{pmatrix} \quad \text{and} \quad \Sigma^* = \begin{pmatrix} Q & \frac{f(x_{k-1})Q}{10} \\ \frac{f(x_{k-1})Q}{10} & \frac{f^2(x_{k-1})Q}{100} + R \end{pmatrix}.$$

For GHQ, we use the 5 point quadrature rule and for JUQ, the three ($n = 1$) sigma points were calculated using $\kappa = 2$. For UPF, the parameters are taken to be the same as that of van der Merwe *et al.* (2000) with $\alpha = 1,\ \beta = 0,\ \kappa = 2$ and $P_0 = 1$. We use, however, systematic resampling scheme while resampling, whereas van der Merwe *et al.* (2000) use residual resampling scheme.

For all methods, the initial distribution $p(x_0)$ is taken to be the true distribution, $\mathcal{N}(0, 5)$ and resampling was done when the effective sample size became less than one-third of the original sample size N. For each method, we first calculate the root mean squared error (RMSE) over $M = 100$ runs for each time point k and then the average (over time) RMSE,

N		LIN	GHQ	JUQ	UPF	EMM
100	RMSE	4.7356	4.5724	4.5647	4.9552	4.6179
	CPU	0.0255	0.0677	0.0617	1.9730	0.0403
	NRS	36.64	31.43	31.35	46.48	31.39
250	RMSE	4.6715	4.4628	4.4989	4.5430	4.4838
	CPU	0.0394	0.1855	0.1684	5.0514	0.0770
	NRS	36.26	32.05	32.29	46.38	32.18
500	RMSE	4.4923	4.4299	4.4567	4.4766	4.4406
	CPU	0.0650	0.5283	0.5008	10.0609	0.1423
	NRS	38.23	32.55	32.43	46.49	32.67
1000	RMSE	4.4310	4.4211	4.4122	4.4491	4.4162
	CPU	0.1352	1.6559	1.6128	20.1678	0.2892
	NRS	39.42	33.29	32.92	46.33	33.23

Table 3.1: Comparison of the performance of different proposal distributions with a polynomial observation model

given by $\frac{1}{100}\sum_{k=1}^{100}\left(\frac{1}{M}\sum_{j=1}^{M}(\hat{x}_k^j - x_k^j)^2\right)^{\frac{1}{2}}$. Here x_k^j is the true (simulated) state for time k in the j-th run and $\hat{x}_k^j$ is the corresponding (point) estimate using a PF method.

Each of these methods is implemented with different Monte Carlo sample sizes $N = 100, 250, 500$ and 1000. In Table 3.1 the average RMSE values are presented. Also reported are the average (over the 100 runs) CPU time, in seconds, to complete a run and the average number of resampling steps (NRS) out of the 100 time steps.

First of all, we see from the table that, as expected, the performances (as measured by RMSE) of all the methods become similar as sample size N increases. This is in conformity with the fact that for any proposal distribution the particle filter converges to the true posterior as $N \to \infty$. The UPF seems to have a considerably higher computational load than the other methods. This can be explained by the fact that one needs to run the unscented Kalman filter for each particle (at each step) to calculate the proposal. Performances of GHQ, JUQ and EMM are more or less similar (which is better than LIN), but the time taken to arrive at the estimate is less for EMM than those for GHQ and JUQ. It appears that the numbers of resampling steps are almost the same for GHQ, JUQ and

EMM, which is slightly better than LIN. So, the extra computational load for GHQ and JUQ relative to our EMM method can be construed as a result of computing the moments numerically. Thus one can conclude that the EMM method is more efficient compared to the other methods as it is computationally less demanding in arriving at the comparable level of efficiency.

3.6.2 Non-polynomial observation model

Let us consider the following model

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + w_k, \quad w_k \sim \mathcal{N}(0,10) \tag{3.6.3}$$

$$y_k = \tan^{-1}(x_k) + v_k, \quad v_k \sim \mathcal{N}(0,1), \quad k = 1,2,\ldots \tag{3.6.4}$$

Once again, a time series data of length 100 was simulated starting with $x_0 \sim \mathcal{N}(0,5)$ and the different particle filters were applied on the observation y_k. Here the exact moments given in (3.3.1) are unknown. For EMM we have considered a 2nd degree Taylor polynomial, as described in section 3.5. The other setup are as in the previous example in section 3.6.1. The performances of different methods are presented in Table 3.2.

Again, comparing the RMSE's we observe that the performances of GHQ, JUQ, UPF and EMM are fairly similar, and they are all better than LIN. But when CPU times are compared, UPF is the worst performer. A very close look reveals that GHQ and JUQ may produce slightly lower RMSE compared to EMM. However, this relative gain is achieved at the expense of high computational load. Thus, considering the trade off between the RMSE and the computational cost, EMM appears to provide a practical and efficient proposal density.

3.6.3 Model with non Gaussian process noise

Here we consider the following model as given in van der Merwe *et al.* (2000):

$$x_{k+1} = 1 + \sin(\omega\pi k) + \phi_1 x_k + w_k \tag{3.6.5}$$

$$y_k = \begin{cases} \phi_2 x_k^2 + v_k, & k \leq 30 \\ \phi_3 x_k - 2 + v_k & k > 30 \end{cases} \tag{3.6.6}$$

N		LIN	GHQ	JUQ	UPF	EMM
100	RMSE	4.1643	4.0875	4.0977	4.1051	4.0936
	CPU	0.0227	0.0686	0.0711	2.0225	0.0567
	NRS	18.13	11.88	12.31	28.76	17.13
250	RMSE	4.1114	4.0592	4.0636	4.0630	4.0669
	CPU	0.0359	0.1977	0.1963	5.1927	0.0752
	NRS	20.79	12.04	12.51	28.70	18.10
500	RMSE	4.1027	4.0416	4.0509	4.0563	4.0524
	CPU	0.0591	0.5930	0.5941	10.3734	0.1102
	NRS	23.40	12.29	12.58	28.61	18.78
1000	RMSE	4.0745	4.0415	4.0426	4.0483	4.0423
	CPU	0.1275	1.6741	1.6936	20.9481	0.2188
	NRS	25.45	12.35	12.64	28.82	19.69

Table 3.2: Comparison of the performance of different proposal distributions with a non-polynomial observation model

where the process noise $w_k \sim Gamma(3, 2)$, $\omega = 0.04$ and $\phi_1 = 0.5$. The observation noise is taken to be $v_k \sim \mathcal{N}(0, 10^{-1})$, $\phi_2 = 0.2$ and $\phi_3 = 0.5$. A time series data of length 60 was simulated starting with initial distribution $p(x_0) \sim \mathcal{N}(0, 5)$. Note that because of the presence of non-Gaussian noise, the methods of LIN, GHQ, JUQ would not work. UPF and EMM methods, however, can be used with this model. So in this example, we compare the performance of EMM to that of UPF. For UPF, the parameters are taken to be the same as that of van der Merwe *et al.* (2000) with $\alpha = 1$, $\beta = 0$, $\kappa = 2$ and $P_0 = 1$ but we use systematic resampling scheme, whereas van der Merwe *et al.* (2000) use residual resampling scheme. For all methods, the initial distribution $p(x_0)$ is taken to be $\mathcal{N}(0, 5)$ and resampling was done when the effective sample size fell below one-third of the original sample size N. Their performances are presented in Table 3.3.

Here also the performances of UPF and EMM are fairly similar, but when CPU times are compared, EMM performs much better. Thus, comparing the overall trade off between RMSE and CPU times, we observe that EMM is more efficient while also relatively easy to implement.

N		**UPF**	**EMM**
200	*RMSE*	0.3429	0.3409
	CPU	2.79	0.0488
	NRS	8.05	7.60
500	*RMSE*	0.3431	0.3449
	CPU	6.2536	0.1100
	NRS	8.37	8.03

Table 3.3: Comparison of the performance of UPF and EMM for a model with non Gaussian process noise

3.7 Comparing different proposals using Kullback-Leibler divergence

It is very common to use RMSE for comparing performances of different particle filtering methods. We have also done the same in the previous section to compare different proposal distributions. RMSE compares the true (synthetic) value of x_k to the mean of the estimated posterior density. One common criticism against the use of RMSE in particle filtering context is that it compares only one aspect (mean) of the estimated distribution with the true one. Thus it neglects all other information which are otherwise available from the posterior represented by the weighted particle cloud. After all, one may have two very differently shaped (estimated) posterior densities with same mean and thus leading to same RMSE value. So for fair comparison, one should compare the complete densities– the true posterior against the estimated posterior. For this purpose, we envisage here the Kullback-Leibler divergence (KLD) measure.

3.7.1 Kullback-Leibler divergence

The KLD is a measure of the difference between two probability distribution (density) functions. The KLD between two probability density functions $p(x)$ and $q(x)$ is given as

$$D(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \ . \tag{3.7.1}$$

If p is not absolutely continuous with respect to q then

$$D(p||q) = +\infty \ . \tag{3.7.2}$$

It is not a proper mathematical measure, since it is not symmetric in its arguments and does not obey the triangle property. Nevertheless, it is widely accepted as a standard measure in Information Theory (Cover and Thomas (1991); Gray (1990)).

3.7.2 Implementation issues

When comparing different PF schemes, one can in principle, use KLD to measure the difference between the true posterior pdf and the estimate obtained by the particular PF scheme. Smaller KLD implies that the estimated posterior obtained by the scheme is closer to the true posterior and hence, the scheme is more appropriate in terms of accuracy. Though the idea is very simple, in practice, the implementation is limited, mainly due to the following reasons:

(a) true posterior pdf of the dynamic state can hardly be obtained analytically.

(b) the filter density is not readily available from particle filter output.

One remedy for (a) could be to run a PF with a very large number of samples and take the output as the true posterior. After all, the PF converges to the true value as the sample size increases to infinity. Regarding (b), the common practice to obtain density from a weighted particle cloud is to use the kernel density estimation method, where one has to specify the bandwidth of the selected kernel. We note here though that this selection is, in general, not trivial. We use the Gaussian kernel density method, where a Gaussian kernel is fitted to each individual particle with bandwidth selected by the 'Rule of Thumb' as given in Silverman (1986). Now, having obtained the densities ($p(x)$ and $q(x)$) in this fashion, one can compute the KLD between them as follows:

$$\begin{aligned} D(p||q) &= \int p(x) \log \tfrac{p(x)}{q(x)} dx \\ &= E_p[\log(p(X))] - E_p[\log(q(X))] \ , \end{aligned}$$

where $X \sim p(\cdot)$ and E_p is the expectation operator with respect to the density $p(x)$. Since it is difficult to get this KL divergence analytically, we resort to a Monte Carlo integration technique where one first draws N_s samples from $p(x)$ and then evaluates the densities $p(x)$ and $q(x)$ at those samples. One can then replace the expected log likelihood by the sample average of the log likelihood such that

$$\begin{aligned} D(p||q) &= E_p[\log(p(X))] - E_p[\log(q(X))] \\ &\simeq \sum_{j=1}^{N_s} \log(p(x_j)) \cdot W_j - \sum_{j=1}^{N_s} \log(q(x_j)) \cdot W_j \end{aligned}$$

where W_j is the weight associated with particle x_j drawn from $p(x)$.

3.7.3 Numerical simulation results

We reconsider the same set up of section 3.6 with system dynamics given by (3.6.1)–(3.6.2). Since the posterior for 'true state' is not analytically available here, we approximate this by the output of a particle filter (with state transition density as proposal) using 20,000 particles. For estimating the posterior density, we use the Gaussian kernel density method. For this purpose, we have used a publicly available Matlab code (Kernel Density Estimation Toolbox for MATLAB (R13)), accessible from the website http://www.ics.uci.edu/∼ihler, (Ihler (2005)). We compare the performance of Gauss Hermite quadrature (GHQ), Julian Ullman quadrature (JUQ) and exact moment method (EMM) by averaging the KLD for each time step over 10 runs. To calculate the KLD for each time step, we draw 1000 samples from the posterior pdf obtained by each method and evaluated the likelihood as mentioned above.

While evaluating this KLD, we encountered numerical instability at some instants. A careful inspection revealed that the kernel density estimator we have used can not properly model the tail distribution. For example, very small values of $q(x_j)$, due to modeling limitations of the estimator, results in $-\infty$ for $E_p[\log(q(x))]$. To avoid this numerical problem, one ad hoc solution is to add noise to the density q so that it does not truncate abruptly. In other words, to replace q by q^* given by

$$q^* = (1 - \epsilon)q + \epsilon Z;$$

where $Z \sim \mathcal{N}(m, \Sigma)$ (i.e. the Gaussian noise with mean m and variance Σ). For the example considered above, we took $\epsilon = 0.005$, $m = 0$ and

PF scheme	KL distance
JUQ	0.8569
GHQ	0.8672
EMM	0.8161

Table 3.4: performance of different proposal densities in terms of KLD

$\sigma = 100$. The KLD averaged over the time for JUQ, GHQ and EMM is given in Table 3.4. We can see that though EMM is slightly better here (smaller KLD), they perform almost equally.

3.8 Concluding Remarks

In this chapter, a new importance function has been proposed which is based on the Gaussian approximation of the conditional distribution of (x_k, y_k), given x_{k-1}, with the first two moments matched exactly to those of the true conditional distribution. To use the proposed method, one needs to know the moments of the system dynamics up to the second order. A specific case in which this is satisfied is when the noise processes are additive Gaussian and the observation equation is polynomial. When the exact moments are not known but the noise processes are additive Gaussian and the observation model is smooth, we use a polynomial approximation of the observation model to derive the importance function. With the help of numerical examples, it has been shown that the proposed EMM method provides a more practical and efficient proposal density considering the trade off between the performance (as measured by RMSE) and the computational load.

Additionally, an effort has been made to compare the performances of the different proposals in terms of Kullback-Leibler divergence. Since one can not get the pointwise density value from the cloud representation directly, we use the (Gaussian) kernel density estimation for this purpose. Using a 'rule of thumb' (Silverman (1986)) for bandwidth selection, the numerical results do not show any difference in the performances of JUQ, GHQ and EMM.
It is however, known that the selection of bandwidth is an important issue in kernel density estimation. While the choice is not at all obvious, it affects the estimate heavily. Furthermore, the kernel density estimation method

is itself computationally very expensive. These limitations in the context of particle filter have motivated us for an alternative density estimation method, which we will explore in the next chapter.

Chapter 4

The Monte Carlo marginal MAP estimator for general state space models

4.1 Introduction

The dynamic estimation problem concerned with estimating the unknown state given a set of measurements is a cornerstone for many practical problems. The complete solution is given by the posterior probability density function, which reflects all knowledge about the current state. From our previous chapters, we recapitulate that for a general nonlinear dynamic system, this posterior is analytically intractable, but can be successfully approximated using PF methods. In this method, the posterior is approximated by a cloud of N weighted particles, whose empirical measure closely approximates the true posterior for large N (Doucet *et al.* (2000); Arulampalam *et al.* (2002); Liu and Chen (1998)). For inference purposes, however, instead of the whole posterior, often a single point estimate is more convenient. One such commonly used point estimate is the minimum mean square error (MMSE) estimate, which happens to be the mean of the posterior density. In PF set up, this can be easily obtained by taking the weighted average of the particles.

Nonetheless, the MMSE estimate has some apparent disadvantages. For example, when the posterior is multimodal, the MMSE estimate may be located in a region (between the modes) where the posterior can have a very small value and thus, producing an unreasonable estimate. In such

a scenario, another point estimate, namely, the maximum *a posteriori* (MAP) estimate, which picks the state that maximizes the posterior, is more relevant. The situation is described in Figure 4.1. In many real life

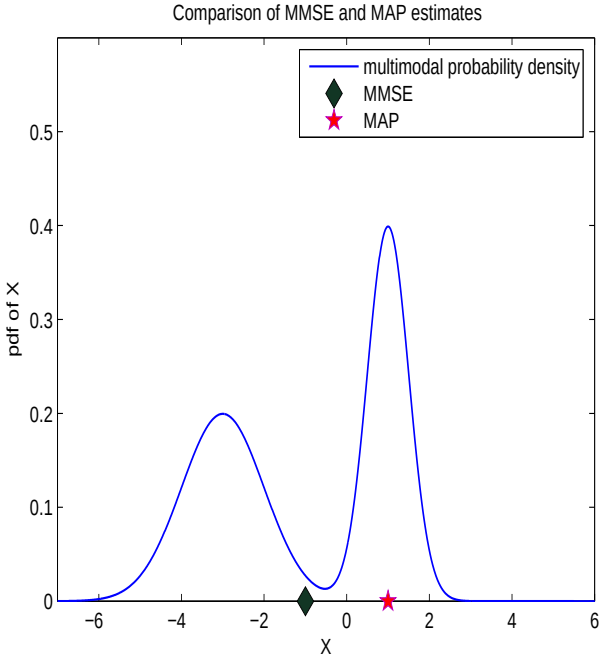

Figure 4.1: Comparison of MMSE and MAP estimates

applications multi modality exhibits itself in a natural manner. Terrain aided navigation is one such example. In this application, the *a posteriori* filtering distribution is defined in terms of the position and velocity of an airplane. The measurements are the terrain heights obtained by taking the difference between the measured absolute altitude (by means of a Inertial Navigation System) and the distance measure to ground (as obtained from Radar Altimeter). This in turn, is compared to the terrain pattern stored in a digital map. However, if the digital map includes different regions which have similar terrain patterns in altitude, a given terrain height can be found at many different positions. This would lead to the multimodality of the posterior. See Nordlund (2008) (pages 14-15) for more details. A snapshot of a particle filter based terrain navigation application in ac-

tion is provided in Figure 4.2. Here, the true trajectory (position) and the corresponding mean estimates as obtained from the particle filter output are shown in green and magenta lines respectively. In this figure, the particle clouds (blue) for the current time step are also shown, which clearly indicate the multimodality of the posterior. Multi modality also appears

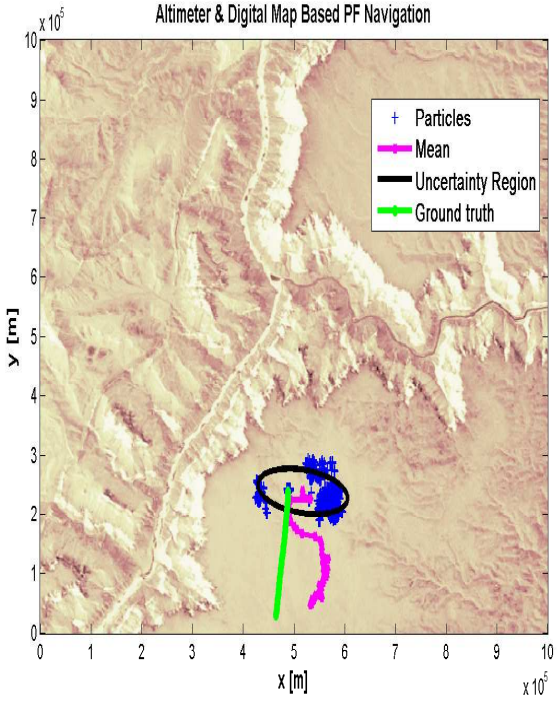

Figure 4.2: Terrain navigation example

in target tracking problems, for example due to data association problems, multiple models of target dynamics (Bar-Shalom and Li (1995); Blom *et al.* (2008)) and the mixed labeling problem in multiple target tracking (Boers and Driessen (2007); Boers *et al.* (2008*a*)). For such applications, extracting the exact object location is very crucial for subsequent strategic decisions and inference purposes. However, the popular MMSE based estimate may potentially lead to a situation where the estimates are far from the actual target positions. This problem is especially apparent in joint multi-target particle filters, see e.g. Boers *et al.* (2008*a*). An illustration

of this is provided in Figure 4.3, showing two targets in a binary sensor network. See Boers *et al.* (2008*b*) for more modeling details.

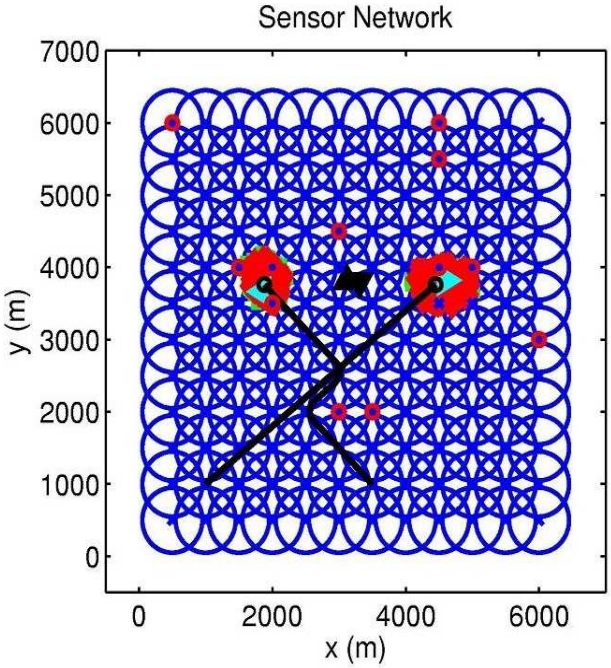

Figure 4.3: Two targets in binary sensor network example

In the example in Figure 4.3 the two targets initially started out well separated, then they move toward each other, remain close to each other for quite some time, before separating again. Moreover, while together, they are also close in terms of the related sensor resolutions. The trajectories of the targets are shown in black lines. The results shown for the current time step are the particle clouds for the two targets (in red and green), the corresponding MMSE estimates (the black triangles) and the true positions of both the targets (the black circles), which are well separated after passing each other as well as the MAP estimates (light blue triangles). We see that after the separation, the two targets cannot be distinguished by the filter (the red and the green clouds are mixed). In this situation, the *a posteriori* probability density is called completely mixed, see Boers *et al.* (2008*a*). Here, the MMSE estimates, which end

up somewhere between the particle clouds, are obviously misleading as both estimates are far away from the actual target positions. The MAP estimates are, on the other hand, much closer to the true target positions.

The examples presented in this section are a motivation for using the MAP estimate instead of the MMSE estimate, at least in certain applications. The rest of the chapter is organized as follows. Following a brief mathematical description of the problem in section 4.2, we will describe two approaches to calculate the filter MAP estimates, based on a running particle filter in section 4.3. Subsequently in section 4.4, We will compare these two MAP estimates both in terms of computational load and performance. There after, we discuss the effect of continuous optimization to extract the filter MAP in section 4.5. In section 4.6, the tweaking of particle populations in constructing the predictive density is studied. Finally, in section 4.7, we extend the MAP estimator to the marginal smoother, which is then applied to estimate the unknown initial state of a dynamic system. For numerical studies, in this chapter, we consider the following two models:

(A) Linear model

$$x_k = \alpha x_{k-1} + w_k, \quad w_k \sim N(0,1) \tag{4.1.1}$$
$$y_k = x_k + v_k, \quad v_k \sim \mathcal{N}(0, 0.01), \quad k = 1, 2, \ldots. \tag{4.1.2}$$

where α is known and the initial distribution is $p(x_0) \sim \mathcal{N}(0, 2)$.

(B) Nonlinear model

$$x_k = \frac{x_{k-1}}{2} + \frac{25 x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2k) + w_k, \quad w_k \sim \mathcal{N}(0, 10) \tag{4.1.3}$$
$$y_k = \frac{x_k^2}{20} + v_k, \quad v_k \sim N(0,1), \quad k = 1, 2, \ldots. \tag{4.1.4}$$

with the initial distribution $p(x_0) \sim \mathcal{N}(0, 5)$.

4.2 Problem Description

We reconsider the state-space model as given in (2.2.1)–(2.2.2). The problem here is to estimate sequentially the current state x_k given all the observations $y_{1:n} \equiv (y_1, y_2, \ldots, y_n)$, $k \leq n$. In this chapter, we will be

concerned with the MAP estimate or equivalently, to estimate the value of x_k that maximizes the posterior density $p(x_k|y_{1:n})$. This can be stated mathematically as

$$x_{k|n}^{MAP} = \arg\max_{x_k} p(x_k|y_{1:n}). \tag{4.2.1}$$

For $k = n$, this is known as filter MAP whereas, for $k < n$, this is called smoothed marginal MAP.

4.3 Filter MAP estimate for a general nonlinear dynamic system

Estimating the MAP involves maximization over the posterior density, which is not readily available either analytically or from the approximation such as particle filter. In the literature, particle with the maximum weight is often taken as the MAP estimate (Zhou *et al.* (2004); Candy (2007); Boers and Driessen (2003)). But, recently it has been shown that the particle with the highest weight does not necessarily represent the most probable state estimate and can actually be far from true MAP (Driessen and Boers (2008*a*); Cappé *et al.* (2007)). Thus, this estimator is not really a fair approximation of the true MAP. Naturally, the crux of the problem lies in constructing the posterior density from the weighted cloud representation of the distribution. We have encountered the same problem in the previous chapter as well when estimating the Kullback-Leibler divergence (section 3.7). As is known, one classic approach is the kernel based method where a kernel is fitted around each particle to approximate the posterior density (Silverman (1986)). This method requires a choice of kernel bandwidth which is not obvious and the method is computationally demanding, which restricts its use in many practical applications, especially those which are online in nature. Moreover, it is worth noticing that the kernel smoothing also increases the variance of the distribution with a value equal to the kernel bandwidth, which is its another drawback.

Recently, Driessen and Boers (2008*a*,*b*) have proposed a new scheme for computing the MAP of the filter distribution, directly from the output of a running particle filter. This method thus avoids the need of bandwidth selection associated with the kernel based method. Furthermore, the approximate MAP estimates obtained through this method are shown to converge to the true MAP estimates. The method is briefly explained

in section 4.3.1. In section 4.3.2, we describe another method which also provides an estimate of x_k (current state) using MAP.

4.3.1 Particle based filter MAP estimate (pf-MAP)

The method has originally been described by Driessen and Boers (2008*a*,*b*). The main advantage of this method is that it can approximate the posterior density not only at particles forming the clouds but at any point. To elucidate the method, consider the same dynamic system as given in equations (2.2.1)-(2.2.2). The MAP estimate of the filtering density at time k is then given by

$$x_{k|k}^{MAP} = \arg\max_{x_k} p(x_k|y_{1:k}). \tag{4.3.1}$$

For a general nonlinear model, analytical expression for the filtering density $p(x_k|y_{1:k})$ can hardly be obtained in closed form. However, using particle filtering technique, one can approximate this posterior distribution by a cloud of N weighted particles as

$$\widehat{P}(dx_k|y_{1:k}) \simeq \sum_{j=1}^{N} w_k^{(j)} \delta_{x_k^{(j)}}(dx_k). \tag{4.3.2}$$

Now using the Bayes' rule, the posterior (filtering) density in equation (4.3.1) can be written as

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \tag{4.3.3}$$

Observing that the denominator is independent of x_k, one can write

$$p(x_k|y_{1:k}) \propto p(y_k|x_k)p(x_k|y_{1:k-1}). \tag{4.3.4}$$

The MAP estimate can thus be expressed as

$$x_k^{MAP} = \arg\max_{x_k} p(y_k|x_k)p(x_k|y_{1:k-1}). \tag{4.3.5}$$

Since the conditional likelihood $p(y_k|x_k)$ is known for each x_k, the main issue for evaluating MAP is the calculation of the predictive density $p(x_k|y_{1:k-1})$. Though not available in closed form, one can use the relation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{4.3.6}$$

and the running particle filter given by (4.3.2) to approximate $p(x_k|y_{1:k-1})$ using Monte Carlo integration as

$$p(x_k|y_{1:k-1}) \approx \sum_j p(x_k|x_{k-1}^{(j)})w_{k-1}^{(j)}. \tag{4.3.7}$$

Substituting (4.3.7) into (4.3.4) we get the posterior density and the MAP estimation is then obtained by finding the global maxima of it. At this point, one can in principle, employ any standard optimization technique to arrive at the MAP estimate. In general, however, this maximization step is nontrivial due to the possible multimodalities arising from the non Gaussian nature of the posterior. An approximate method for the MAP estimate is as follows.

With the view that the cloud of particles $\{x_k^{(i)}\}_{i=1}^N$in a running particle filter constitutes an adaptive discretization of the state space at time k, one can approximately locate the MAP by first evaluating equation (4.3.4) at the predicted particle $\{x_k^{(i)}\}_{i=1}^N$ and finally selecting the particle with the highest density. This leads to the approximate particle based MAP estimate (pf-MAP) as

$$x_k^{MAP} = \arg\max_{x_k^{(i)}} p(y_k|x_k^{(i)}) \sum_j p(x_k^{(i)}|x_{k-1}^{(j)})w_{k-1}^{(j)}. \tag{4.3.8}$$

It is clear from equation (4.3.7) that the predictive density can be seen to be a weighted mixture of state transition densities. It is worthwhile to note here that in certain situations, this might not work well. For example, when the variance of the process noise is very low such that the mixture components are well separated. In such a situation, a support point which lies between two such mixture components can have unreasonably low predictive density, although the true likelihood may be high. As a consequence. this will result in a very low posterior density. This situation may arise, for example, when the parameter is treated as augmented state with small artificial process noise and starting with a wide initial distribution. Here, if the particles sampled from the initial distribution are away from the true parameter, due to the small process noise variance, the particles can not be propagated (in time) to the true neighborhood of the parameter and as a result, the posterior would be very low around the true neighborhood (in the limit, as the variance of the artificial process noise goes to zero, the continuous nature of this predictive density breaks down to the sum of Dirac-delta functions).

For the theoretical convergence properties of this estimator, the reader is referred to Driessen and Boers (2008*b*). One should note that for each time step, the memory requirement of this MAP estimator is $O(N)$ and the computational complexity is $O(N^2)$. This complexity may possibly be reduced using the method suggested by Klaas *et al.* (2006). We do not discuss this any further.

4.3.2 End Point Viterbi-Godsill MAP (EP-VGM)

In the recent past, Godsill *et al.* (2001) have developed a method for estimating the MAP sequence in nonlinear non-Gaussian dynamic models using the Viterbi algorithm. At each time k, the method uses the particle cloud representation as an adaptive discretization of the state space and then employs the Viterbi algorithm on this discretized state space to find the MAP estimate of the sequence $x_{0:k}$. In target tracking problem, this corresponds to providing (new) MAP estimates of the whole trajectory of the target at every time point k (using the new observation). Clearly, the last element of this sequence can be viewed as an estimate of the current state. Subsequently, we call this estimate as end point Viterbi-Godsill MAP (EP-VGM) estimate of x_k. Actually in this method, the last element is obtained first and then using backward recursion, the other elements of the MAP sequence are obtained. For details, see Godsill *et al.* (2001).

4.4 Comparing pf-MAP with EP-VGM

In this section we compare the behaviors of EP-VGM and pf-MAP as an estimate of x_k. It is to be noted though that the estimates obtained using EP-VGM and the pf-MAP are not necessarily the same. While the Viterbi-Godsill algorithm aims at maximizing $p(x_{0:k}|y_{1:k})$, pf-MAP aims at maximizing $p(x_k|y_{1:k})$. The two estimates would be, at least in principle, the same when the system dynamics are linear-Gaussian, as $p(x_{0:k}|y_{1:k})$ is then multivariate Gaussian. However, when there is significant nonlinearity and/or non-Gaussianity in the model, such that the joint posterior $p(x_{0:k}|y_{1:k})$ is non-Gaussian, the output of these estimators can be quite different. For an illustration, please see Boers *et al.* (2009). Subsequently, we investigate their behaviors through numerical simulations. In this context, we emphasize that the calculation of EP-VGM does not require the backtracking step of the Viterbi algorithm as we do not seek the whole

sequence. Consequently, the CPU time for EP-VGM does not involve any backtracking step.

For comparison purpose, we first start with the linear Gaussian model (i.e. model (A)). For this model, we take $\alpha = 1$. Here, the true MAP for the current state can be extracted using Kalman filter. The root mean square error (RMSE) estimate with respect to true MAP over 20 Monte Carlo runs with 200 time steps along with the average CPU time (in second) are shown in Table 4.1 and Table 4.2. The average CPU time reported here

	pf-MAP	EP-VGM
Sample	RMSE	RMSE
100	0.007459	0.007430
200	0.006604	0.006653
400	0.006202	0.006224
1000	0.005948	0.006034

Table 4.1: Linear Gaussian model: RMSE

	pf-MAP	EP-VGM
Sample	avg. CPU	avg. CPU
100	0.0085	0.4765
200	0.0265	1.7461
400	0.1008	7.8359
1000	0.4890	49.9367

Table 4.2: Linear Gaussian model: average CPU time

does not include the cost of computation common for both the methods as we have used the same running particle filter for each Monte Carlo run. It is evident that for the same number of particles, RMSE's for both the methods behave similarly while the average CPU time for pf-MAP is substantially less.

Next we consider the nonlinear model(B). Since for this example, the true MAP can not be obtained analytically, we compare them as an estimator of the current state. The RMSE estimate is done with respect to true (synthetic) state over 20 Monte Carlo runs with 200 time steps. The results are shown in Table 4.3 and Table 4.4:

	pf-MAP	EP-VGM
Sample	RMSE	RMSE
100	5.2964	5.5667
250	4.9707	5.2567
500	4.8530	5.4184
1000	4.4659	5.2433

Table 4.3: Nonlinear model: RMSE

	pf-MAP	EP-VGM
Sample	avg. CPU	avg. CPU
100	3.515e-05	0.0033
250	1.836e-04	0.0194
500	6.054e-04	0.0851
1000	0.0022	0.3457

Table 4.4: Nonlinear model: average CPU time

We observe that in terms of RMSE estimate, pf-MAP performs slightly better as an estimate of current state while it is also computationally much cheaper.

4.5 Improvement over the estimated pf-MAP

In section 4.3.1, the pf-MAP is estimated by comparing the density values at the current particles and choosing the one with the highest density. We note, however that the predictive density $p(x_k|y_{1:k-1})$ can be approximated, as given by (4.3.7), at any point x_k. The same is then true for the posterior density $p(x_k|y_{1:k})$ by virtue of (4.3.4). Consequently, we can use different optimization methods such as Genetic Algorithm or any gradient based continuous optimization method to maximize the posterior density. For subsequent numerical experiments in this section, we use the unconstrained optimization function 'fmincon' of MATLAB with default parameter setting. Here, the pf-MAP as obtained in section 4.3.1, is taken as initial starting point for the MAP estimate using gradient based optimization technique (subsequently referred to as 'grad-MAP').

We consider the nonlinear model as given in equations (4.1.3)-(4.1.4).

For the running particle filter, we use $N = 500$ particle, $T = 200$ and state transition density as proposal. For a typical run, the results are shown in Figure 4.4 through Figure 4.6.

We observe that the gradient-MAP obtained with the pf-MAP as a start-

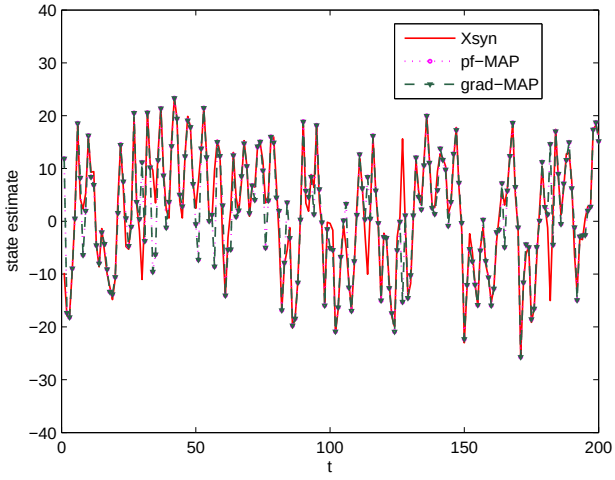

Figure 4.4: Nonlinear model: simulated state (Xsyn), pf-MAP and grad-MAP over time

ing point has slight marginal edge over the pf-MAP in terms of estimation efficiency while, it is computationally heavier due to the further overload of continuous optimization method.

Another suggestion to improve the overall performance would be to use only a subset of particles in the maximization step of finding pf-MAP of section 4.3.1 and with this (crude) estimate as starting point, obtain the grad-MAP. This may result in a high reduction of computational load, with the level of estimation efficiency expected to remain similar. Furthermore, this approach can be combined with reducing the number of particles in constructing the predictive density (as discussed in the next section). However, these suggestions are not pursued here further.

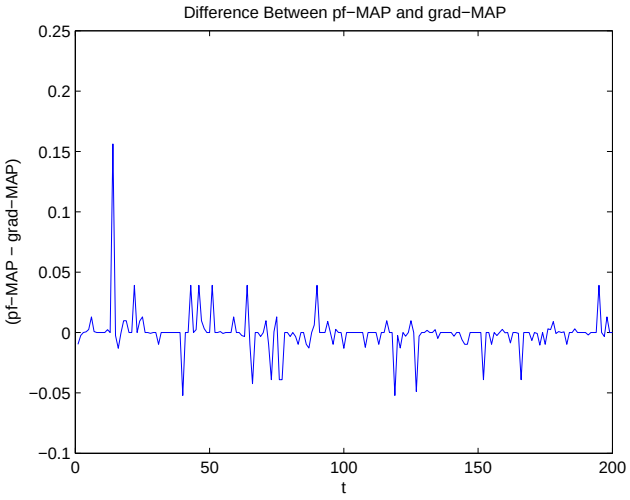

Figure 4.5: Nonlinear model: difference between pf-MAP and grad-MAP over time

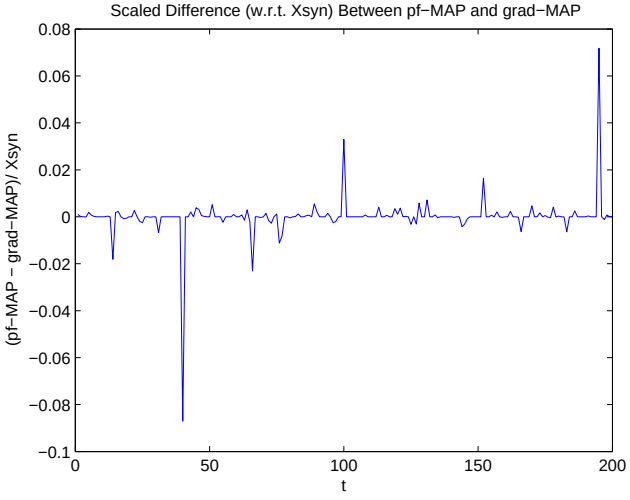

Figure 4.6: Relative difference between pf-MAP and grad-MAP (w.r.t. simulated state (Xsyn))

4.6 Particle tweaking effect on predictive density

While evaluating the pf-MAP, we have additional flexibility to explore the effect of tweaking the number of particles used in constructing the predictive density. The predictive density is approximately obtained from the running particle filter using the Monte Carlo integration as given in equation (4.3.7). For this Monte Carlo integration, instead of using the whole cloud, one can use a subset of $\{x_{k-1}^{(j)}, w_{k-1}^{(j)}\}_{j=1}^{N}$ to reduce the computational cost. The reduced set of particle (N_R) can be obtained from the original cloud, for example, through a resampling step. We subsequently compare the tweaking effect numerically in terms of average CPU time (in second) and RMSE estimate over 20 Monte Carlo runs with 200 time steps. For the running particle filter, the state transition density is chosen as the proposal. For the linear model (with $\alpha = 0.5$), the RMSE is estimated with respect to the true MAP extracted from Kalman filter (with $p(x_0) \sim \mathcal{N}(0, 2)$) and the result is shown in Table 4.5. We observe that the

Original size (N)	Tweaking size (N_R)	RMSE	Avg. CPU
500	500	0.0216	15.48
	250	0.0216	11.49
	125	0.0216	09.55
1000	1000	0.0153	66.47
	500	0.0153	45.13
	250	0.0153	37.82

Table 4.5: Tweaking : linear model

tweaking leads to huge computational savings while it does not affect the accuracy in terms of RMSE estimate. To investigate the effect further, we next consider the nonlinear model. Since, in this case, the true MAP is not known analytically, the RMSE is estimated with respect to true (synthetic) state over 20 Monte Carlo runs with 200 time steps. The result is shown in Table 4.6.

Here also, we observe that by reducing the number of particles in constructing predictive density, the average CPU time falls drastically while the RMSE does not change so much.

Original size (N)	Tweaking size (N_R)	RMSE	Avg. CPU
1000	1000	5.1922	78.10
	500	5.1608	54.26
	250	5.2115	46.39
	100	5.3503	39.81

Table 4.6: Tweaking : nonlinear model

In conclusion, it seems that the particle tweaking in predictive density evaluation can lead to a substantial improvement in computational saving at a little compromise on RMSE value.

4.7 Particle based smoothed marginal MAP estimator (ps-MAP)

So far we have considered the different aspects of the filter MAP estimation using the weighted particles. In this section, our main objective is to extend the MAP estimation idea to marginal smoothing. We again consider the same dynamic system as given by (2.2.1) and (2.2.2). Now, the problem can be mathematically posed as finding

$$x_{t|T}^{MAP} = \arg\max_{x_t} p(x_t|y_{1:T}) \tag{4.7.1}$$

where $t < T$.

Our starting point is that there already exists a (weighted) particle cloud for the marginal smoother. Based on these weighted cloud representation, we calculate the marginal smoothing density using similar ideas as in pf-MAP. From the literature, we find that "forward-backward smoothing" is the conventional approach to generate the particle clouds of a marginal smoother. Subsequently, we derive the smoothed marginal MAP from it. As an application we use this marginal MAP to find the initial condition. Recently "generalized two-filter smoothing" has been used to generate the particle cloud for the marginal smoother. In this case as well, we explain how to extract the smoothed marginal MAP from the particle clouds.

4.7.1 Forward-Backward Smoothing

The marginal smoother can be obtained using a forward- backward smoother (Kitagawa (1987)) as

$$p(x_t|y_{1:T}) = p(x_t|y_{1:t}) \int \frac{p(x_{t+1}|y_{1:T})p(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} dx_{t+1}, \tag{4.7.2}$$

where, $p(x_t|y_{1:t})$ and $p(x_{t+1}|y_{1:t})$ are the filtering density and one step ahead predictive density respectively, at time t. Thus, starting with $p(x_T|y_{1:T})$, $p(x_t|y_{1:T})$ can be recursively obtained from $p(x_{t+1}|y_{1:T})$. Using the above recursion, the marginal smoothing distribution can now be approximated by the weighted particle cloud as described in (Briers *et al.* (2004); Hürzeler and Künsch (1998)). Here, one starts with the forward filtering pass for computing the filtered distribution at each time step using the particle filter as

$$\widehat{P}(dx_t|y_{1:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta_{x_t^{(i)}}(dx_t). \tag{4.7.3}$$

Then one performs the backward smoothing pass as given by (4.7.2) to approximate the smoothing distribution

$$\widehat{P}(dx_t|y_{1:T}) = \sum_{i=1}^{N} \omega_{t|T}^{(i)} \delta_{x_t^{(i)}}(dx_t), \tag{4.7.4}$$

where the smoothing weights are obtained through the following backward recursion:

$$\omega_{t|T}^{(i)} = \omega_t^{(i)} \sum_{j=1}^{N} [\omega_{t+1|T}^{(j)} \frac{p(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^{N} p(x_{t+1}^{(j)}|x_t^{(k)})\omega_t^{(k)}}] \tag{4.7.5}$$

with $\omega_{T|T}^{(i)} = \omega_T^{(i)}$. It is important to note that the forward-backward smoother keeps the same particle support as used in filtering step and re-weights the particles to obtain the approximated particle based smoothed distribution. Thus, success of this method crucially hinges on the filtered distribution having supports where the smoothed distribution is significant.

To obtain the smoothed marginal MAP, one needs the posterior density $p(x_t|y_{1:T})$ from the above cloud representation. Here, we proceed as

follows. Using the Bayes' rule, one can write the one step ahead predictive density in equation (4.7.2) as

$$p(x_{t+1}|y_{1:t}) = \frac{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})}{p(y_{t+1}|x_{t+1})}. \tag{4.7.6}$$

Then equation (4.7.2) becomes

$$\begin{aligned} p(x_t|y_{1:T}) &= p(x_t|y_{1:t}) \int \frac{p(x_{t+1}|y_{1:T})p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})} dx_{t+1} \\ &= \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \int \left[\frac{p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})} \right] p(x_{t+1}|y_{1:T}) dx_{t+1} \\ &\approx \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \int \left[\frac{p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})} \right] \widehat{P}(dx_{t+1}|y_{1:T}). \end{aligned}$$

Making use of the particle representation of $\widehat{P}(dx_t|y_{1:T})$, given by (4.7.4), and subsequently approximating the above integration by a Monte Carlo integration method, one obtains

$$p(x_t|y_{1:T}) \approx \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \sum_{j=1}^{N} \left[\frac{p(x_{t+1}^{(j)}|x_t)p(y_{t+1}|x_{t+1}^{(j)})}{p(x_{t+1}^{(j)}|y_{1:t+1})} \right] \omega_{t+1|T}^{(j)}. \tag{4.7.7}$$

Further approximating the filtered density $p(x_{t+1}|y_{1:t+1})$ from the running particle filter (Driessen and Boers (2008*a*)) as

$$p(x_{t+1}|y_{1:t+1}) \approx \frac{p(y_{t+1}|x_{t+1}) \sum_k p(x_{t+1}|x_t^{(k)}) w_t^{(k)}}{p(y_{t+1}|y_{1:t})} \tag{4.7.8}$$

we can rewrite equation (4.7.7) as

$$p(x_t|y_{1:T}) \approx p(x_t|y_{1:t}) \sum_{j=1}^{N} \left[\frac{p(x_{t+1}^{(j)}|x_t)}{\sum_{k=1}^{N} p(x_{t+1}^{(j)}|x_t^{(k)}) \omega_t^{(k)}} \right] \omega_{t+1|T}^{(j)}. \tag{4.7.9}$$

The MAP estimate of the marginal smoothing density, $p(x_t|y_{1:T})$ can then be obtained by finding the location of its global maximum. This maximization can be performed using different optimization methods. As in

the case of pf-MAP, here, as well, we maximize along the particles. This leads to the approximate particle based MAP estimate as

$$x_{t|T}^{MAP} \approx \arg\max_{x_t^{(i)}} p(x_t^{(i)}|y_{1:t}) \sum_{j=1}^{N} \left[\frac{p(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^{N} p(x_{t+1}^{(j)}|x_t^{(k)})\omega_t^{(k)}} \right] \omega_{t+1|T}^{(j)}, \quad (4.7.10)$$

where N is the number of particles used at each time step. By using equation (4.7.5), the estimator can be further simplified to

$$x_{t|T}^{MAP} = \arg\max_{x_t^{(i)}} p(x_t^{(i)}|y_{1:t}) \frac{\omega_{t|T}^{(i)}}{\omega_t^{(i)}}, \quad (4.7.11)$$

where the filtered density $p(x_t|y_{1:t})$ at the particle cloud $\{x_t^{(i)}\}_{i=1}^{N}$ can be evaluated during the forward filtering step (Driessen and Boers (2008*a*)) as

$$p(x_t^{(i)}|y_{1:t}) \approx \frac{p(y_t|x_t^{(i)}) \sum_j p(x_t^{(i)}|x_{t-1}^{(j)}) w_{t-1}^{(j)}}{p(y_t|y_{1:t-1})}. \quad (4.7.12)$$

Since $p(y_t|y_{1:t-1})$ in equation (4.7.12) is independent of $x_t^{(i)}$, to obtain $x_{t|T}^{MAP}$, one can replace $p(x_t^{(i)}|y_{1:t})$ in equation (4.7.11) by the un-normalized filtered density

$$q(x_t^{(i)}|y_{1:t}) = p(y_t|x_t^{(i)}) \sum_j p(x_t^{(i)}|x_{t-1}^{(j)}) w_{t-1}^{(j)}. \quad (4.7.13)$$

We note here that a numerical problem may arise in evaluating equation (4.7.11) if the filtered weights attached to some particles are very small. This may happen when the "particle degeneracy" occurs. This problem can be effectively addressed using a combination of efficient importance proposal (Doucet *et al.* (2000); Guo *et al.* (2005); Saha *et al.* (2009*b*)) along with resampling steps. The algorithm for the smoothed marginal MAP proposed above is given below.

Algorithm :

- `Given observation` $y_{1:T}$`,`
  `For` $i = 1, .., N$, `where` N `is the number of particles`

Forward Filtering step

- Assume $p(x_0)$, draw $x_0^{(i)}$ from $p(x_0)$, set $\omega_0^{(i)} = \frac{1}{N}$.
- Run Particle Filter to generate and store $\{x_t^{(i)}, \omega_t^{(i)}\}$ for $t = 0, ..., T$
- Evaluate (un-normalized) filtered pdf for $t = 1, ..., T$, at cloud points i

$$q(x_t^{(i)}|y_{1:t}) = p(y_t|x_t^{(i)}) \sum_j p(x_t^{(i)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)}$$

starting with $q(x_0^{(i)}) = p(x_0^{(i)})$ and store

Backward Smoothing step

- Set $\omega_{T|T}^{(i)} = \omega_T^{(i)}$
- For $t = T-1, ..., 0$ evaluate the smoother importance weights as

$$\omega_{t|T}^{(i)} = \omega_t^{(i)} \sum_{j=1}^{N} [\omega_{t+1|T}^{(j)} \frac{p(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^{N} p(x_{t+1}^{(j)}|x_t^{(k)})\omega_t^{(k)}}]$$

- Evaluate the approximate smoothed marginal MAP as

$$x_{t|T}^{MAP} = \arg\max_{x_t^{(i)}} q(x_t^{(i)}|y_{1:t}) \frac{\omega_{t|T}^{(i)}}{\omega_t^{(i)}}$$

The memory requirement of this marginal MAP smoother is $O(N)$ and the computational complexity is $O(N^2)$.

4.7.1.1 Estimation of (unknown) initial condition

We first consider the linear Gaussian model as given by (4.1.1)-(4.1.2) with $\alpha = 0.8$, but, the initial state x_0 is assumed to be unknown (constant). The synthetic data $\{x_k, y_k\}_{k=0:500}$ is generated starting with $x_0^* = 10$. To

estimate the unknown initial state x_0, we start with initial prior $p(x_0) \sim U[0,\ 20]$ where $U[a,\ b]$ denotes uniform probability density function with lower bound a and upper bound b respectively. We use "efficient proposal" as given in Doucet *et al.* (2000) in the forward filtering step with particle sample size $N = 500$. The estimate of the initial unknown state is given by the particle based MAP of $p(x_0|y_{0:T})$. We repeat this MAP state estimate for 30 Monte Carlo runs. The mean and variance of the estimator are shown in Table 4.7. The result shows that the smoothed initial density peaks around the true initial state, even though we have started with a pretty wide uniform initial prior. We also plot for a particular realization, the (backward) evolution of the marginal smoother estimates (i.e. mean and the MAP) for the first 10 time steps and the un-normalized filtered and smoothed probability density functions (pdfs) of x_0 in Figure 4.7 and Figure 4.8 respectively. As expected, the mean and MAP are almost similar and the smoothed density is more concentrated than the filtered density around the true value 10. Next, we consider the nonlinear time series

$Mean(x_{0\|500}^{MAP})$	$Var(x_{0\|500}^{MAP})$
9.9726	0.0915

Table 4.7: Mean and Variance of estimated initial state

model as given by (4.1.3) and (4.1.4). The synthetic data $\{x_k, y_k\}_{k=0:500}$ is generated starting with $x_0^* = 10$. As in the previous case, we start with initial prior $p(x_0) \sim U[0,\ 20]$. For this nonlinear problem, we use the "Exact Moment matching (EMM) proposal" as given in Saha *et al.* (2006) during forward filtering step with particle sample size $N = 500$. The estimate of the initial unknown state is given by the particle based MAP of $p(x_0|y_{0:T})$. We repeat this MAP state estimate for 30 Monte Carlo runs. The mean and variance of the estimator are shown in Table 4.8. The result in Table 4.8 is really remarkable as we can see by comparing with Table 4.7. Even for highly nonlinear model as considered above and with wide uniform initial prior, the result is almost as good as in linear case. Of course the variance is somewhat larger, but that is to be expected given the highly nonlinear nature of the problem.

It is also interesting to study the behavior of the smoother when the initial distribution is supported on a larger interval. Starting with $p(x_0) \sim U[-40,\ 40]$, the (backward) evolution of the marginal smoother es-

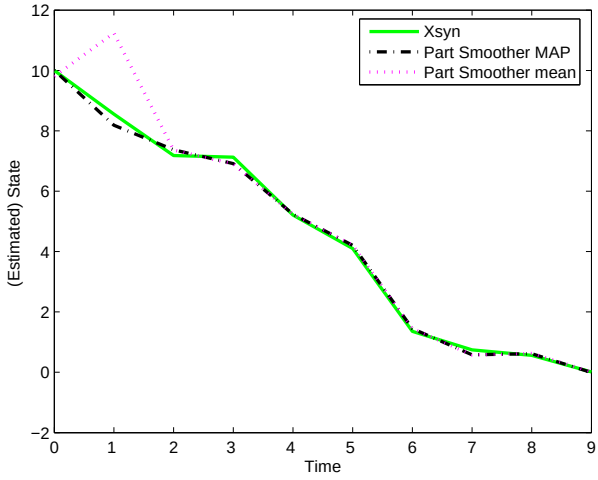

Figure 4.7: Simulated state (Xsyn), MAP and mean of the marginal smoothing posterior for the first 10 time steps

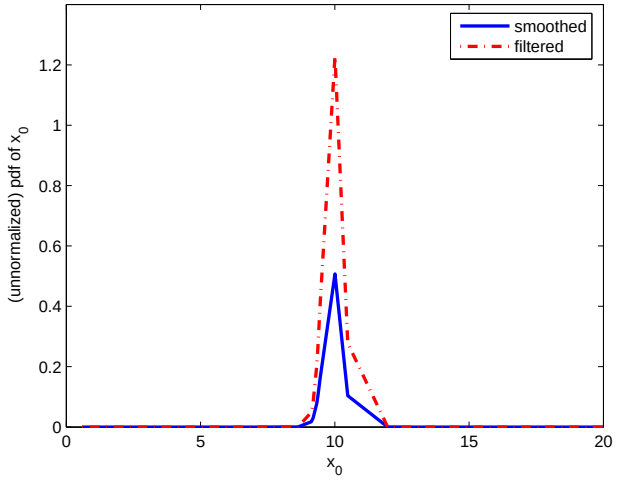

Figure 4.8: Filtered and smoothed probability density functions for the initial state x_0

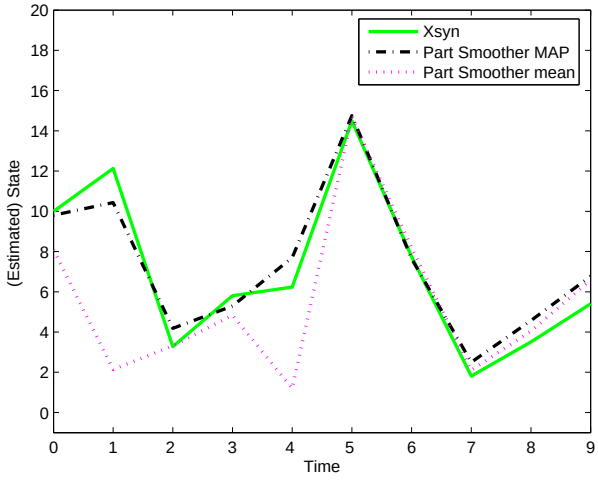

Figure 4.9: Simulated state (Xsyn), MAP and mean of the marginal smoothed posterior for the first 10 time steps

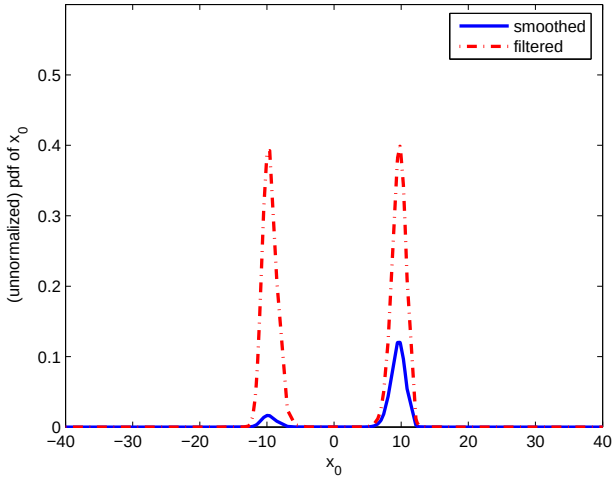

Figure 4.10: Filtered and smoothed probability density functions for the initial state x_0

$Mean(x_{0\|500}^{MAP})$	$Var(x_{0\|500}^{MAP})$
9.7165	0.9236

Table 4.8: Mean and Variance of estimated initial state

timates (i.e. mean and the MAP) for the first 10 time steps for a particular realization are shown in Figure 4.9 while the corresponding un-normalized filtered and smoothed pdfs for x_0 are shown in Figure 4.10. It is interesting to note that the smoothed pdf of the initial state is bimodal (the smaller peak is near -10). Although the dominant mode is very close to the true initial state, $x_0^* = 10$, the contribution from the weaker mode, shifts the smoothed mean away from x_0^* (as seen from Figure 4.9, the smoothed mean is near 8 here). This further strengthens the justification of using MAP in such scenario.

4.7.2 Two-Filter Smoothing

One shortcoming of the forward-backward smoother is its reliance on the support points (particles) generated during the forward filtering pass. To circumvent this problem, two-filter smoother has been envisaged in the literature (Briers *et al.* (2004); Kitagawa (1996); Isard and Blake (1998)), where one combines samples from particle filter in the forward direction with those from a so called "backward information filter" to produce the (weighted) cloud representation of $p(x_t|y_{1:T})$. We describe, in this section, how the smoothed marginal MAP can be obtained from the particle cloud generated by the generalized two-filter smoother. We start with a brief description of how two filter particle smoother is obtained. For this, we follow Briers *et al.* (2004).

In the two-filter smoother framework, the so-called backward information filter $p(y_{t:T}|x_t)$ is calculated sequentially from $p(y_{t+1:T}|x_{t+1})$ as

$$p(y_{t:T}|x_t) = p(y_t|x_t) \int p(x_{t+1}|x_t) p(y_{t+1:T}|x_{t+1}) dx_{t+1}. \tag{4.7.14}$$

As noted by Briers *et al.* (2004), $p(y_{t:T}|x_t)$ is not a probability density function in x_t and actually, its integral over x_t may not even be finite. The smoothing algorithm in (Kitagawa (1996); Isard and Blake (1998)) assumes implicitly that $\int p(y_{t:T}|x_t)dx_t < \infty$. However, if this assumption does not hold, SMC based methods, which can only approximate finite

measures, will not work anymore. To avoid this, "generalized two-filter smoothing" has been proposed by Briers *et al.* (2004), where the smoothing distributions are computed through a combination of forward filter and an auxiliary probability distribution $\widetilde{p}(x_t|y_{t:T})$ in argument x_t. This auxiliary density is defined through a sequence of artificial distributions $\gamma_t(x_t)$ as

$$\widetilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_{t:T}|x_t).$$

It then follows from (4.7.14) that

$$\widetilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_t|x_t)\int p(x_{t+1}|x_t)\frac{\widetilde{p}(x_{t+1}|y_{t+1:T})}{\gamma_{t+1}(x_{t+1})}dx_{t+1}. \quad (4.7.15)$$

This in turn, is used to generate recursively the weighted particle representation of the backward information filter

$$\widetilde{p}(x_t|y_{t:T}) \simeq \sum_{k=1}^{N}\delta(x_t - \widetilde{x}_t^{(k)})\widetilde{\omega}_t^{(k)}. \quad (4.7.16)$$

The marginal smoother $p(x_t|y_{1:T})$ is then computed by combining the outputs of the forward filter and the backward information filter as

$$\begin{aligned} p(x_t|y_{1:T}) &\propto p(x_t|y_{1:t-1})p(y_{t:T}|x_t) \\ &= \left(\int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}\right)\left(\frac{\widetilde{p}(x_t|y_{t:T})}{\gamma_t(x_t)}\right). \end{aligned} \quad (4.7.17)$$

Evaluating the integral in (4.7.17) by Monte Carlo integration using the forward filter cloud $(x_{t-1}^{(j)}, \omega_{t-1}^{(j)})$ one obtains

$$p(x_t|y_{1:T}) \propto \left(\sum_{j=1}^{N} p(x_t|x_{t-1}^{(j)})\omega_{t-1}^{(j)}\right)\left(\frac{\widetilde{p}(x_t|y_{t:T})}{\gamma_t(x_t)}\right). \quad (4.7.18)$$

Finally, the particle cloud representation is obtained using the cloud $(\widetilde{x}_t^{(k)}, \widetilde{\omega}_t^{(k)})$ from the backward filter:

$$p(x_t|y_{1:T}) \simeq \sum_{k=1}^{N}\delta(x_t - \widetilde{x}_t^{(k)})\widetilde{\omega}_{t|T}^{(k)} \quad (4.7.19)$$

where

$$\widetilde{\omega}_{t|T}^{(k)} \propto \frac{\widetilde{\omega}_t^{(k)}}{\gamma_t(\widetilde{x}_t^{(k)})}\sum_{j=1}^{N} p(\widetilde{x}_t^{(k)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)}. \quad (4.7.20)$$

Thus, in essence the particles from the forward filter are used to re-weight those from the backward filter so that they represent the marginal smoother distribution. We refer the readers to the original article by Briers *et al.* (2004) for more details.

Now we describe how to derive the smoothing density from the particle smoother obtained as above. Note that using (4.7.16) one can rewrite equation (4.7.15) as

$$\widetilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_t|x_t)\sum_{k=1}^{N}\frac{p(\widetilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\widetilde{x}_{t+1}^{(k)})}\widetilde{\omega}_{t+1}^{(k)}. \tag{4.7.21}$$

It then follows from (4.7.18) that

$$p(x_t|y_{1:T}) \propto \left(\sum_{j=1}^{N} p(x_t|x_{t-1}^{(j)})\omega_{t-1}^{(j)}\right)\left(p(y_t|x_t)\sum_{k=1}^{N}\frac{p(\widetilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\widetilde{x}_{t+1}^{(k)})}\widetilde{\omega}_{t+1}^{(k)}\right). \tag{4.7.22}$$

The required smoothed marginal MAP can now be obtained by maximizing the unnormalized smoothing density, given by the right hand side of equation (4.7.22). Furthermore, when this maximization is done along the particles $\widetilde{x}_t^{(i)}$, we have

$$\begin{aligned}
&p(\widetilde{x}_t^{(i)}|y_{1:T}) \\
&\propto \left(\sum_{j=1}^{N} p(\widetilde{x}_t^{(i)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)}\right)\left(p(y_t|\widetilde{x}_t^{(i)})\sum_{k=1}^{N}\frac{p(\widetilde{x}_{t+1}^{(k)}|\widetilde{x}_t^{(i)})}{\gamma_{t+1}(\widetilde{x}_{t+1}^{(k)})}\widetilde{\omega}_{t+1}^{(k)}\right) \\
&= \left(\frac{1}{\gamma_t(\widetilde{x}_t^{(i)})}\sum_{j=1}^{N} p(\widetilde{x}_t^{(i)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)}\right)\left(\gamma_t(\widetilde{x}_t^{(i)})p(y_t|\widetilde{x}_t^{(i)})\sum_{k=1}^{N}\frac{p(\widetilde{x}_{t+1}^{(k)}|\widetilde{x}_t^{(i)})}{\gamma_{t+1}(\widetilde{x}_{t+1}^{(k)})}\widetilde{\omega}_{t+1}^{(k)}\right).
\end{aligned}$$

From equations (4.7.20) and (4.7.21) this reduces to

$$p(\widetilde{x}_t^{(i)}|y_{1:T}) \propto \left(\frac{\widetilde{\omega}_{t|T}^{(i)}}{\widetilde{\omega}_t^{(i)}}\right)\left(\widetilde{p}(\widetilde{x}_t^{(i)}|y_{t:T})\right). \tag{4.7.23}$$

Hence, the required MAP can be obtained as

$$x_{t|T}^{MAP} = \arg\max_{\widetilde{x}_t^{(i)}} \widetilde{p}(\widetilde{x}_t^{(i)}|y_{t:T})\frac{\widetilde{\omega}_{t|T}^{(i)}}{\widetilde{\omega}_t^{(i)}}, \tag{4.7.24}$$

where $\widetilde{p}(\widetilde{x}_t^{(i)}|y_{1:T})$ is evaluated using equation (4.7.21).

4.8 Concluding Remarks

The posterior of a nonlinear non Gaussian dynamic system can be successfully constructed using particle filtering. The MMSE is a popular point estimate and can easily be obtained from this posterior. However, for certain practical applications like target tracking, where multi modality of posterior is very common, MMSE does not always provide a reasonable estimate. In such situations, MAP estimator can serve as a good alternative to MMSE. Recently a new particle based MAP estimator corresponding to a general state space model has been derived (Driessen and Boers (2008*a*)). In this chapter, we study this particle filter based MAP estimates. A comparison of this with the so called Viterbi-Godsill MAP sequence estimator for estimating the current state shows the superiority of the former. Exploiting the fact that the existing method provides the posterior density $p(x_t|y_{1:t})$ at any support point x_t, we have employed gradient based optimization method, which does not restrict itself to the particles when finding the MAP. We have, however, found that the gradient based method demands more computational load while estimation efficiency remaining almost the same. Some suggestions were made for improvement, though they have not been pursued. It has been shown numerically that by tweaking the number of particles during the evaluation of predictive density, one can reduce the computational load substantially without compromising much on RMSE value.

Finally, we have extended the idea of filter MAP to develop smoothed marginal MAP based on a particle representation of the smoother distribution $p(x_t|y_{1:T})$. We derive a very simple formula for the smoothed marginal MAP when either forward-backward smoother or generalized two-filter smoother is used to generate the particle cloud. The smoothed marginal MAP is applied to estimate the unknown initial condition of a dynamic system. We observe that the estimation works quite well even in nonlinear setting.

Chapter 5

Parameter estimation using particle filtering/smoothing

5.1 Introduction

Nonlinear filtering has been a focus of interest in statistical and engineering community for more than 30 years (Ristic *et al.* (2004)). As discussed in the previous chapters, starting with Gordon's seminal paper (Gordon *et al.* (1993)), particle filtering (PF) method has been getting increasing attention for solving such nonlinear and/or non Gaussian estimation problems. The popularity of PF stems from its generality in the sense that it does not require any ad-hoc approximation of the dynamic models and it is relatively easy to implement. In this method, the posterior is approximated by a cloud of $N(N \gg 1)$ weighted particles, whose empirical measure closely approximates the true posterior for large N(Arulampalam *et al.* (2002)). Standard PF algorithms assume the perfect knowledge of the static parameters of the underlying model whereas, in many practical situations, often those model parameters of the dynamic system are not known *a priori* and their estimation is also of interest. However, in this context, standard PF fails and it is necessary to rely on more sophisticated algorithms. Generic solutions using the particles based method for parameter estimation, which are useful for any model are still limited in performance, thus opening up the possibility of further research in this direction.

Among the existing approaches, the classical remedy is to augment the parameter as additional state with artificial dynamics and then tak-

ing filtered estimates of the additional state vector as the estimates of parameters. The artificial evolution, however, in effect, renders the fixed parameter into a slowly varying one (Liu and West (2001)). As a result, the variance of the estimate of the parameter increases with time. Another proposed scheme is to marginalize the static parameters out of the posterior either analytically (Djurić and Miguez (2002)) or by Monte Carlo procedures (Storvik (2002)). However, such methods are strictly model dependent. Andrieu *et al.* (2005) also proposed an online estimation method for static parameters with the assumption that the state space models are stationary and ergodic. Among others, the particle based maximum likelihood (ML) estimator has also been developed recently. In this framework, when the number of parameters is small, one can consider the direct particle approximation of the log likelihood evaluated on a grid of values of parameters (Olsson and Rydén (2008)). However, when the dimension of parameter vector is large, optimizing the log likelihood through a grid based approximation becomes unwieldy and calls for a more structured and efficient optimization strategy like gradient based optimization or the Expectation-Maximization (EM) (see Cappé *et al.* (2007); Kantas *et al.* (2009) and the references there). Although estimates based on ML are asymptotically optimal, particle based implementation is rather complicated and the convergence is known to require a substantial amount of data (Doucet and Tadić (2003)). We do not get into the EM algorithm any further. For an excellent discussion on the EM algorithm, see the recent article by Ninness (2009).

In this chapter, we propose some new particle filtering/smoothing based schemes for estimating the parameter of a general state space model. In all these schemes, we avoid any effect of artificial noise on the (final) estimate of the parameter. These schemes are described in the subsequent sections (section 5.2–5.5). Finally we end this chapter with a conclusion.

5.2 Augmented state space formulation without any parameter dynamics

We Consider the following dynamic system:

$$x_{k+1} = f(x_k, w_{k+1}; \theta), \tag{5.2.1}$$

$$y_k = h(x_k, v_k), \qquad k = 0, 1, \ldots \tag{5.2.2}$$

where θ is a fixed unknown parameter, (x_k) are the unobservable states with (known) initial prior density $p(x_0)$ and (y_k) are the observations. The process noises (w_k) are assumed to be independent of the measurement noises (v_k). Here, we treat the unknown parameter as an additional state component, but without any dynamic evolution. The augmented state space can be written as

$$\begin{aligned} x_{k+1} &= f(x_k, \theta_k, w_{k+1}) && (5.2.3)\\ \theta_{k+1} &= \theta_k && (5.2.4)\\ y_k &= h(x_k, v_k), \qquad k = 0, 1, \ldots && (5.2.5) \end{aligned}$$

Now, using notation $X_{k+1} = [x_{k+1}\ \theta_{k+1}]^t$ and $W_{k+1} = [w_{k+1}\ 0]^t$, the above model can be rewritten as

$$X_{k+1} = g(X_k, W_{k+1})$$

$$y_k = h(X_k, v_k).$$

For this method, we assume that the interval within which the parameter lies, is known, which may be realistic in many practical situations and is also considered in other estimation techniques (Doucet and Tadić (2003)). In this method, the particles for each parameter are initially selected arbitrarily from any assumed (possibly uniform) distribution, whose support covers the known interval for the parameter. At each subsequent time step, the particles representing the parameter for the initial time step, are resampled using simple random sampling and are then propagated to the next time step. The particles representing the true state are resampled using systematic resampling, only when the effective sample size falls below the threshold, which is two-third of the samples used. In each time step, the individual parameter is estimated by weighted average of particles representing the parameter, where the weights are the importance weights of the corresponding particles representing the state. A known limitation of this method is that, in absence of any evolution for parameter, the exploration of the parameter space is limited to the initial cloud generated for the parameter. We demonstrate the proposed method on the benchmark time series model (Gordon *et al.* (1993)) given by

$$\begin{aligned} x_k &= f(x_{k-1}; \theta_1, \theta_2) + w_k, && (5.2.6)\\ y_k &= \frac{x_k^2}{20} + v_k, && (5.2.7) \end{aligned}$$

where $f(x_{k-1};\theta_1,\theta_2) = \theta_1 x_{k-1} + \frac{\theta_2 x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k)$ and θ_1 and θ_2 are unknown parameters. We assume $p(x_0) \sim \mathcal{N}(0,5)$, $w_k \sim \mathcal{N}(0,10)$ and $v_k \sim \mathcal{N}(0,1)$, which are mutually independent. For simulation, we use $T = 300$, $N = 2000$, $\theta_1(0) \sim U[0,1]$, $\theta_2(0) \sim U[20,30]$, where $U[L,U]$ denotes uniform distribution with lower bound L and upper bound U. The estimate of the parameters θ_1 and θ_2 are shown in Figure 5.1 and Figure 5.2 respectively. We see that although the estimates are oscillating

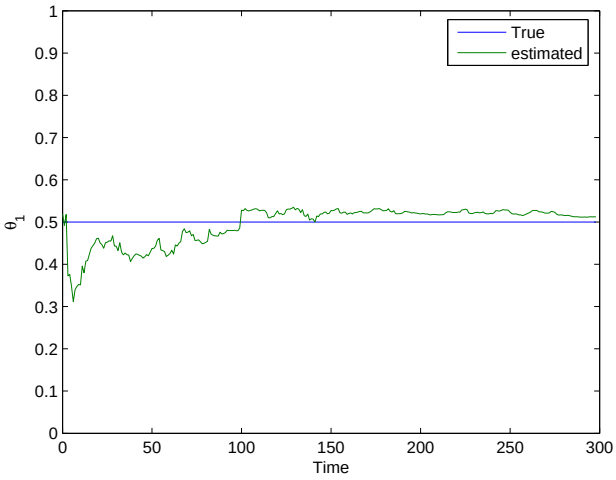

Figure 5.1: Estimate of θ_1

around the true values, they are quite reasonable. Furthermore, the behavior of the estimated parameters are similar to the case of adding the roughening noise to the parameter values as described in section 5.4 below.

Subsequently, we apply this method for estimating the stochastic volatility and the model parameters from the observed stock data. This is a notoriously difficult problem and a major challenge in mathematical finance. For this purpose, we consider here the Heston model with jumps (Bates model), which is highly popular in mathematical finance (Aihara *et al.* (2008)). For the original Heston model (without jump), see Aihara *et al.* (2009).
We have already outlined the particle filtering implementation of Bates model in appendix B for the purpose of estimating the volatility. In this

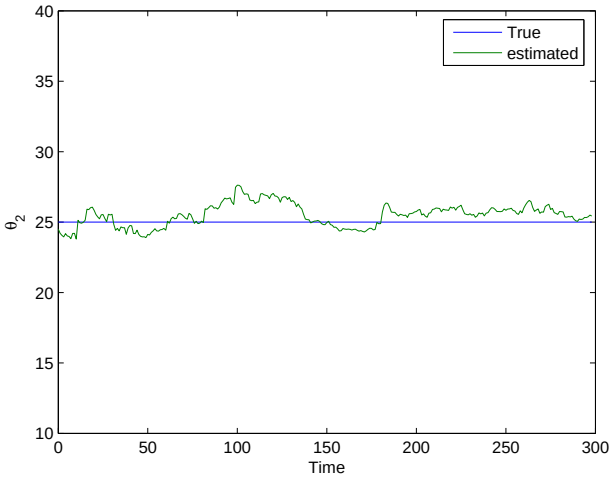

Figure 5.2: Estimate of θ_2

section, we want also to identify the parameters contained in the system model. For this, we construct the augmented state $z_k = (v_k, \alpha)$ where

$$\alpha = [\kappa\ \theta\ \xi\ \mu_S\ \rho\ \lambda\ \mu_J\ \sigma_J].$$

To perform the particle filter for z_k we assume that each component of α is independent and uniformly distributed with known upper and lower bounds, and is independent of the initial distribution of v_0, which is taken as Gaussian. Hence we can apply the particle filter algorithm outlined above to z_k-process. Noting that the state α is time independent and if the parameter value $\alpha^{(i)}$ is not updated, we encounter the so called degeneration problem. To partially mitigate this deficiency, we use here the simple random resampling for each parameter and apply the systematic resampling for the state v_k. Repeating this resampling for every parameter at each time step, we observe that the estimated parameters do not degenerate.
We next performed the simulation studies. In the subsequent simulations, resampling for the state $v_{(.)}$ is done whenever the effective sample size (as defined by equation (2.3.15)) falls below two-third of the sample size used. To check the algorithm proposed here, the stock price and volatility

process are simulated using the following values of the parameters:

$$\kappa = 3.0,\ \theta = 0.1,\ \mu = 0.1,\ \rho = -0.2,$$
$$\xi = 0.4,\ \lambda = 5.0,\ \mu_J = -0.1,\ \sigma_J = 0.2.$$

The simulated volatility and the log price $y(t)$ are shown in Figure 5.3 and Figure 5.4 respectively. In this simulation studies, we set $\Delta t = 0.001$ and use 2000 particles.

For the unknown parameters, we set

$$\kappa \sim U[1, 10],\quad \theta \sim U[0.05, 0.5],\quad \mu \sim U[0.05, 0.3],$$
$$\xi \sim U[0.01, 0.91],\ \rho \sim U[-0.8, 0]$$
$$\lambda \sim U[0, 7],\ \mu_J \sim U[-0.2, 0],\ \sigma_J \sim U[0, 0.4].$$

We also set

$$v_0 \sim N(0.25, 0.02^2).$$

The true and estimated volatility are demonstrated in Figure 5.5. The

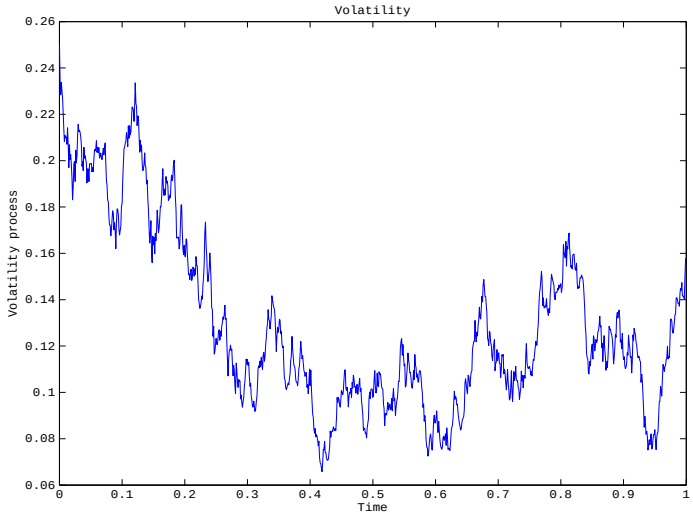

Figure 5.3: Simulated volatility

estimates of unknown parameters are shown in Figures 5.6–5.13. Some of the parameters show distinct bias. It is known that the estimation of the

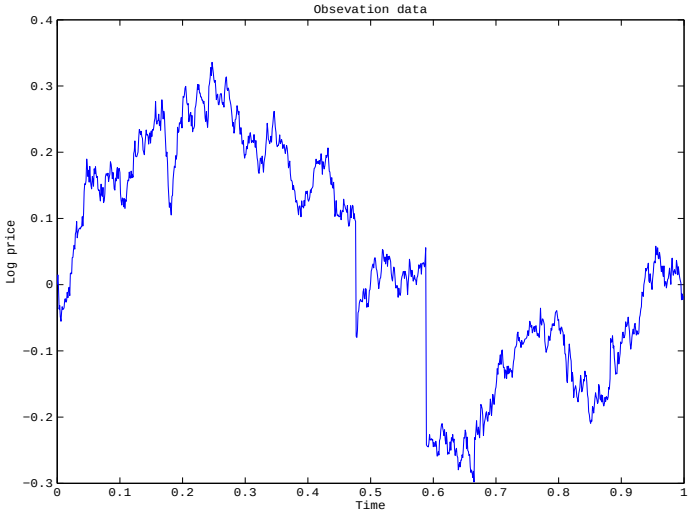

Figure 5.4: Observation data (log price)

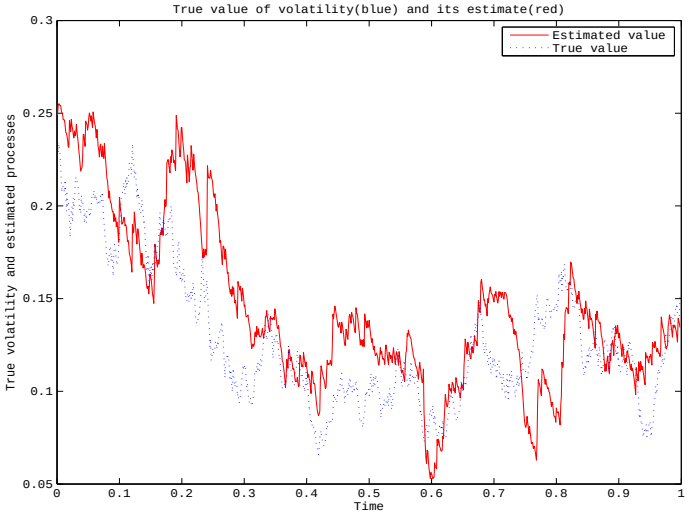

Figure 5.5: True and estimated volatility processes

parameter κ is very difficult. In this example, the total number of parameters is quite large. As a result, with merely 2000 samples as used here, the effective exploration of the state space in region where the joint probability of z_k is high, becomes difficult. Increasing the number of samples will possibly result in better estimate. This needs further study.

The above method is very heuristic in its approach. The estimation exercise seems to show mixed results. We proceed further with other methods.

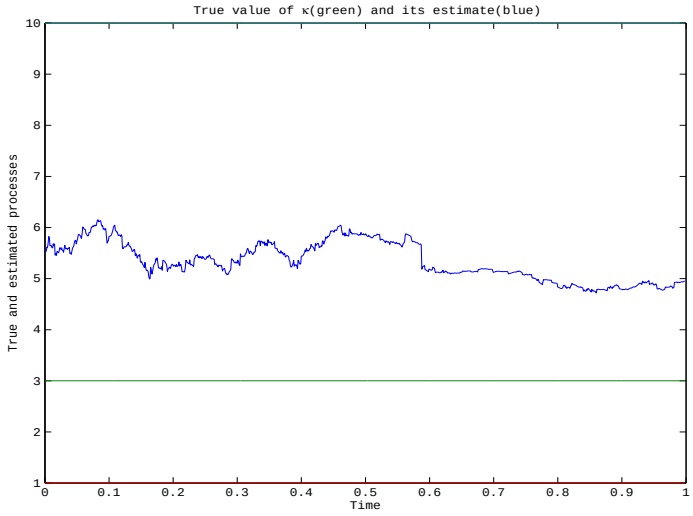

Figure 5.6: True and estimated κ

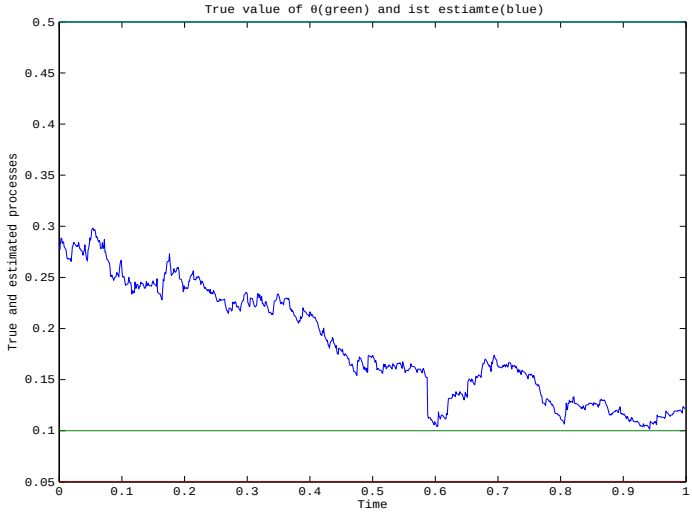

Figure 5.7: True and estimated θ

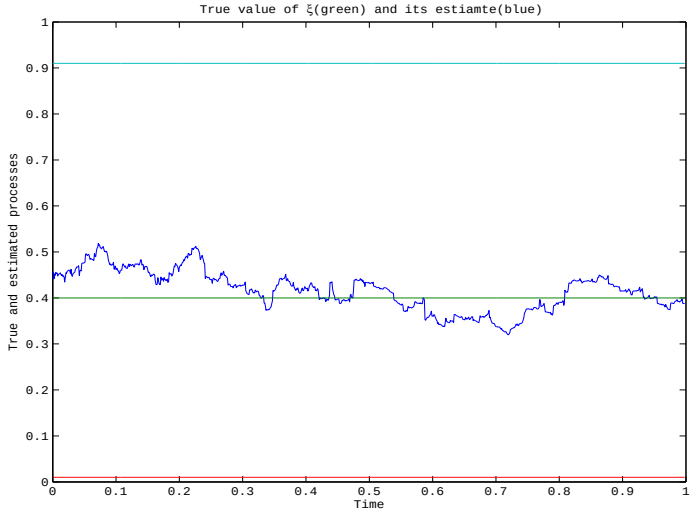

Figure 5.8: True and estimated ξ

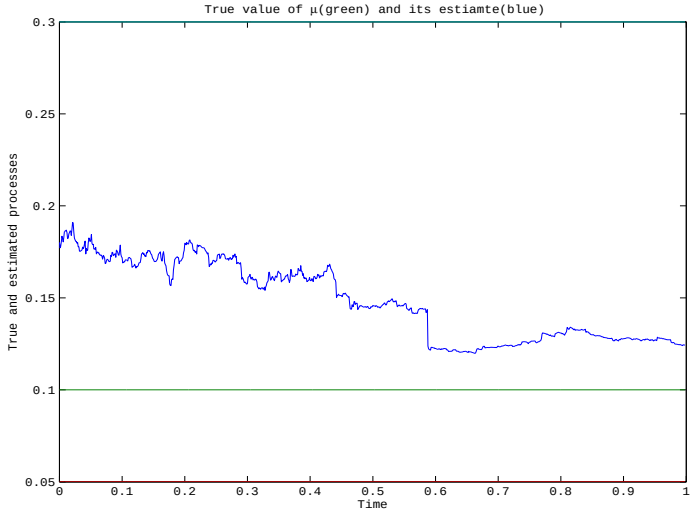

Figure 5.9: True and estimated μ

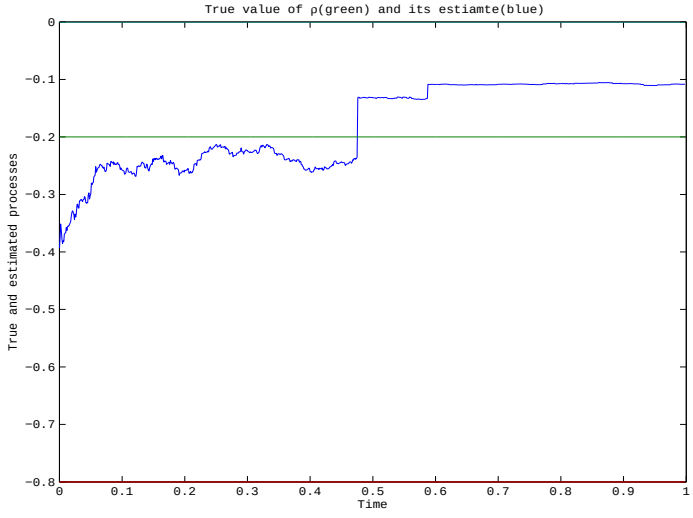

Figure 5.10: True and estimated ρ

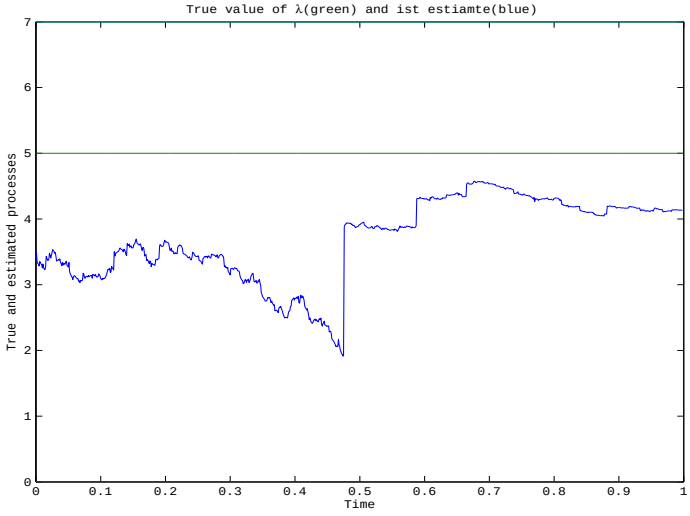

Figure 5.11: True and estimated λ

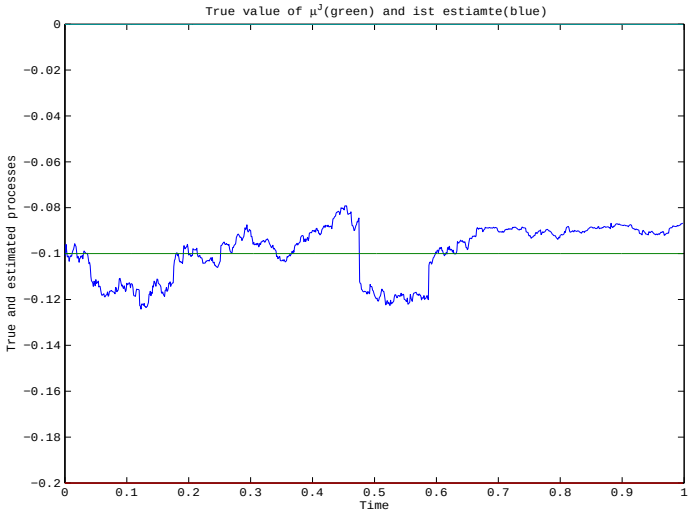

Figure 5.12: True and estimated μ_J

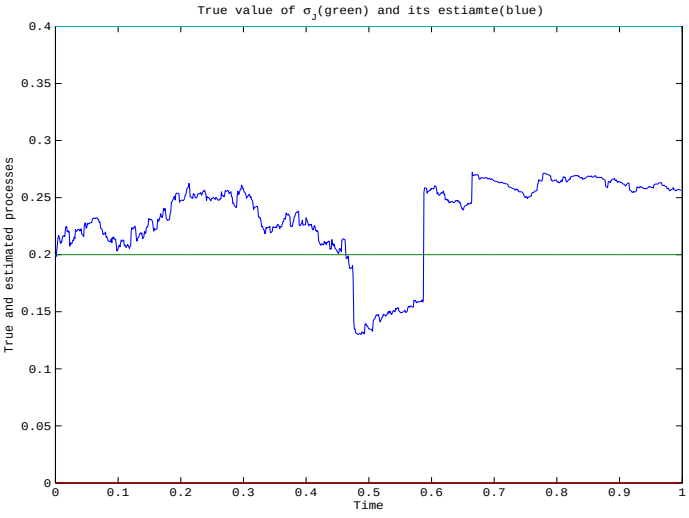

Figure 5.13: True and estimated σ_J

5.3 Parameter estimation using smoothed marginal MAP

As is well known, one of the common approaches of estimating a parameter in a state-space model is to augment the parameter as an extra state with small artificial dynamics and then take the filtered estimate as the estimate of the parameter. Since in this augmented framework, the initial augmented state vector is free of any artificial dynamics (i.e. not corrupted by any artificial noise), we consider the marginal smoother of the initial augmented state to be the estimate of the true (fixed) parameter (Saha *et al.* (2008*b*)). It is expected that as more and more observations are available, the smoothed estimate would converge to the true parameter value. We start here with the same dynamic system as given in equations (5.2.1)–(5.2.2). Next we augment the state space by treating the parameter as additional state. Note that the dimension of the state increases by the numbers of parameters. Now the augmented state space can be written as

$$\begin{aligned} x_{k+1} &= f(x_k, \theta_k, w_{k+1}) && (5.3.1)\\ \theta_{k+1} &= \theta_k + \eta_{k+1} && (5.3.2)\\ y_k &= h(x_k, v_k), \qquad k = 0, 1, \ldots && (5.3.3) \end{aligned}$$

with $\theta_0 = \theta$, which is unknown here, η_{k+1} is the artificial noise driving the additional state (parameter) in the augmented framework. Now, using the notation $X_{k+1} = [x_{k+1}\ \theta_{k+1}]^t$ and $W_{k+1} = [w_{k+1}\ \eta_{k+1}]^t$, the above model can be rewritten as

$$\begin{aligned} X_{k+1} &= g(X_k, W_{k+1})\\ y_k &= h(X_k, v_k). \end{aligned}$$

Then we can estimate the initial state vector X_0 using smoothed marginal MAP as developed in section 4.7 of the previous chapter. The corresponding estimate for the augmented state θ_0 would be taken as the estimate of the parameter. This is a batch method of estimating the parameter. We illustrate this approach by the following two numerical examples below. For all these examples, we use Forward-Backward marginal MAP smoother. We begin with a linear example:

$$x_k = \theta x_{k-1} + w_k \tag{5.3.4}$$

$$y_k = x_k + v_k \tag{5.3.5}$$

with $w_k \sim \mathcal{N}(0,1)$, $v_k \sim \mathcal{N}(0,0.1)$. w_k is independent of v_k and the the (unknown) true parameter is given by $\theta^* = 0.5$. We take $\eta_k \sim \mathcal{N}(0,0.0025)$. Note that θ_0 is independent of x_0. With $p(x_0) \sim \mathcal{N}(0,5)$, we started with $p(\theta_0) \sim U[-5,\ 5]$. We use $N = 1000$ particles and state transition density, $p(x_k|x_{k-1})$ as our proposal during forward filtering step. Although, for stability one should consider θ here to be between $(-1,1)$, one may not know the stability zone in unknown system. It seems that the particle filter adjusts itself automatically. The mean and the standard deviation of the estimator of θ over 30 Monte Carlo runs are shown in Table 5.1 below. Although the assumption of uniform initial prior is radically

True parameter	$Mean(\theta^{MAP}_{0\|500})$	$Std(\theta^{MAP}_{0\|500})$
0.5	0.422	0.265

Table 5.1: True parameter, mean and standard deviation of the estimated parameter

different from the knowledge of exact initial condition (parameter), we see the parameter estimate to be quite good. Next we consider the following nonlinear example:

$$x_k = \frac{x_{k-1}}{2} + \frac{\theta x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + w_k, \tag{5.3.6}$$

$$y_k = \frac{x_k^2}{20} + v_k, \tag{5.3.7}$$

where $w_k \sim \mathcal{N}(0,10)$ and $v_k \sim \mathcal{N}(0,1)$, which are independent of each other. The true parameter is $\theta^* = 25$. With $p(x_0) \sim \mathcal{N}(0,5)$, we started with $p(\theta_0) \sim U[-50,\ 50]$. We use $N = 1000$ particles and state transition density as proposal during forward filtering step. We set $\eta_k \sim \mathcal{N}(0,5)$. The estimate of θ for 30 Monte Carlo runs is shown in Table 5.2. As remarked after Table 5.1, we see the same pattern in a nonlinear problem as well. We observed that this estimation procedure works quite well even in nonlinear cases. However, the computational burden with the growing memory requirement is a major stumbling block here. Additionally, when the number of parameters is large, the dimension of X_{k+1} also increases and the effective exploration of the state space in region where the joint probability of X_{k+1} is high, becomes difficult with a finite number of (relatively small) samples.

True parameter	$Mean(\theta_{0\|500}^{MAP})$	$Std(\theta_{0\|500}^{MAP})$
25.0	27.259	1.241

Table 5.2: True parameter, mean and standard deviation of the estimated parameter

5.4 Parameter estimation using the marginalized particle filter method

In many practical situations, the underlying general state space contains a linear substructure, subject to Gaussian noise. In those cases, the linear sub structure can be marginalized out and one may use the standard Kalman filter for this part and particle filter on the remaining part of the state space model. These two algorithms can be combined into a single algorithm, known as marginalized particle filter (Schön *et al.* (2005)), also known as Rao-Blackwellized particle filter (Doucet *et al.* (2000); Andrieu and Doucet (2002)) and mixture Kalman filter (Chen and Liu (2000)). The resulting posterior distribution is obtained as an optimal Gaussian mixture approximation to the filtering distribution. Since in this algorithm we are applying particle filter to a lower dimensional state space and the optimal filter to the linear substructure, the resulting estimate is expected to be more accurate than the situation where particle filter is used for the entire state space model. For a discussion about the possible class of models where marginalization can be used and the detailed algorithm, one may refer to Schön *et al.* (2005) and Karlsson *et al.* (2005). The potential use of marginalized particle filter for parameter estimation, where the state space model is linear in the parameter, has been considered by Schön and Gustafsson (2003). These authors have considered the evolution of the parameter by adding an artificial noise (also known as "roughening noise"'), whose variance is reduced at each time step. We note that the state space model here is linear in the parameter, where a Kalman filter is actually used. Since the Kalman filter can be applied even without a process noise, at least in theory, we can get rid of this "artificial noise" in the parameter estimation and this is the line we pursue in this section. We first compare our approach to Schön and Gustafsson (2003) for the Chaos example given there and subsequently, we estimate the parameter of the benchmark time series model.

5.4.1 A "Chaos" example

The chaos model as considered in Schön and Gustafsson (2003) is as follows:

$$x_{k+1} = (1 - x_k)x_k\theta + w_k \tag{5.4.1}$$
$$y_k = x_k + v_k \tag{5.4.2}$$

where x_k is the state variable, y_k is the measurement and θ is the unknown parameter. Here w_k is the process noise and v_k is the measurement noise, which are independent of each other. By treating θ as an additional state, Schön and Gustafsson (2003) have rewritten the above model (equation (5.4.1)–(5.4.2)) as

$$x_{k+1} = A_k(x_k)\theta_k + w_k + w_k^x \tag{5.4.3}$$
$$\theta_{k+1} = \theta_k + w_k^\theta \tag{5.4.4}$$
$$y_k = h_k(x_k) + v_k \tag{5.4.5}$$

where $A_k(x_k) = (1 - x_k)x_k$, $h_k(x_k) = x_k$. $w_k^x \sim \mathcal{N}(0, Q_k^{w,x})$ and $w_k^\theta \sim \mathcal{N}(0, Q_k^{w,\theta})$ are the roughening noises introduced by them in this model. The variances of these roughening noises are reduced at each time step according to Gustafsson and Hriljac (2003). Furthermore, process noise w_k is assumed to be zero, $v_k \sim \mathcal{N}(0, R_k)$. The true value of θ is 3.92. The initial state is $x_0 \sim \mathcal{N}(0, 1)$. For the particle filter, state transition density is used as proposal. For all simulations, $Q_0^{w,x} = 10^{-2}$, $R_k = 10^{-5}$ and $N = 150$ (particles), if not stated explicitly otherwise. In the following, we first validate our Matlab code by reproducing the parameter estimate with roughening noise ($Q_0^{w,\theta} = 10^{-2}$) in the parameter space (section 5.4.1.1) and then in section 5.4.1.2, remove the roughening noise in the parameter space, i.e. $Q_0^{w,\theta} = 0$.

5.4.1.1 Parameter estimation with roughening noise in the parameter space

The initial guess is the same as in Schön and Gustafsson (2003), which is taken as $\theta_{0|-1} \sim \mathcal{N}(3.83, 0.04)$. The estimate of θ is shown in Figure 5.14. Noting that the mean of this initial guess of θ is too close to the true value, we next started with $\theta_{0|-1} \sim \mathcal{N}(3.5, 0.04)$, and, in this case also, we obtained good estimate, which is shown in Figure 5.15. For this example, it is not obvious how to select $Q_0^{w,x}, Q_0^{w,\theta}$ and whether R_k plays

any role for this selection. We next set $R_k = 10^{-2}$ and checked whether the lowering of this signal to noise ratio has any effect in parameter estimate. As is observed from Figures 5.16–5.17, the estimate is oscillating around the true value and the convergence is comparatively slower in this case.

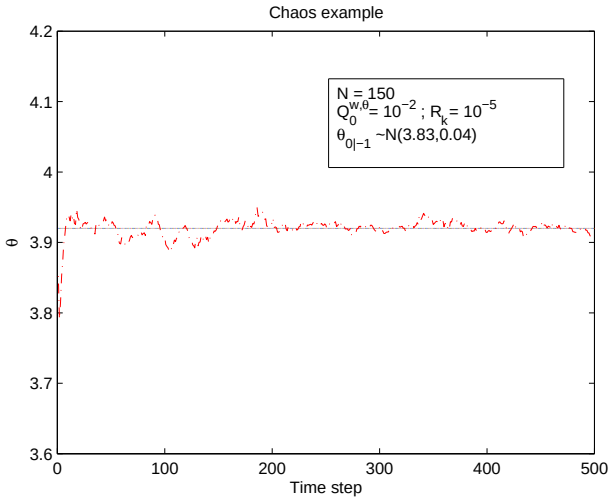

Figure 5.14: Estimate of θ with $Q_0^{w,\theta} = 10^{-2}$

5.4.1.2 Parameter estimation without roughening noise in the parameter space

We again start with the same set up as in Schön and Gustafsson (2003), except that $Q_0^{w,\theta} = 0$. The estimate is shown in Figure 5.18, which is converging to the true value. Subsequently, we have repeated the estimation with $\theta_{0|-1} \sim \mathcal{N}(3.5, 0.04)$ and from Figure 5.19, we observed the convergence in this case as well. However, when starting with $R_k = 10^{-2}$, we found that, to explore the parameter set initially, we need to set a higher variance for $\theta_{0|-1}$. With $\theta_{0|-1} \sim \mathcal{N}(3.83, 0.5)$, the corresponding estimate is shown in Figure 5.20. Thus, we observe that, even without adding roughening noise to the parameter space in marginalized method, one can obtain good parameter estimates. We next use this marginalized method for estimating the parameter of the benchmark time series model.

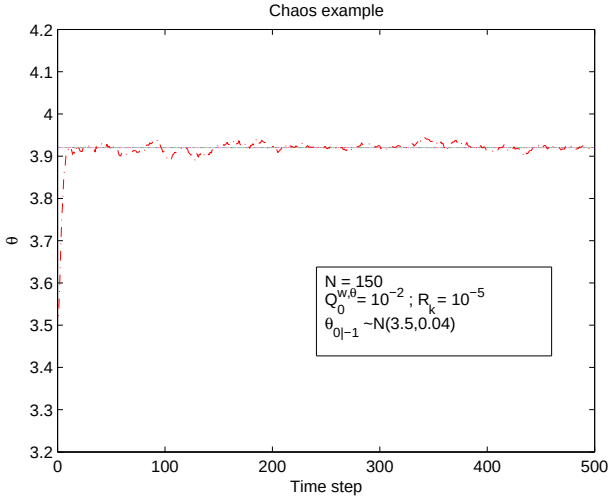

Figure 5.15: Estimate of θ with $Q_0^{w,\theta} = 10^{-2}$

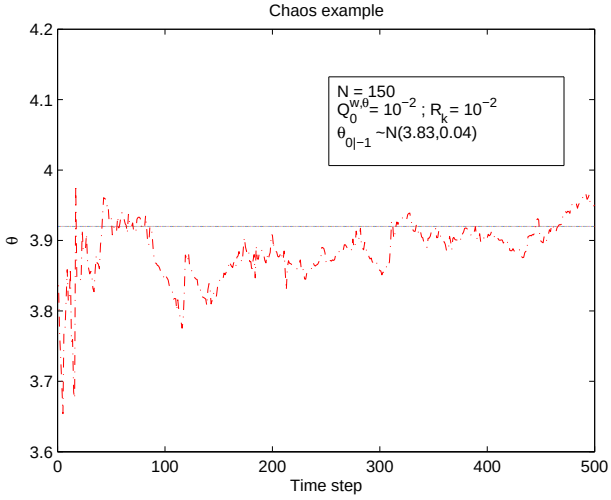

Figure 5.16: Estimate of θ with $Q_0^{w,\theta} = 10^{-2}$

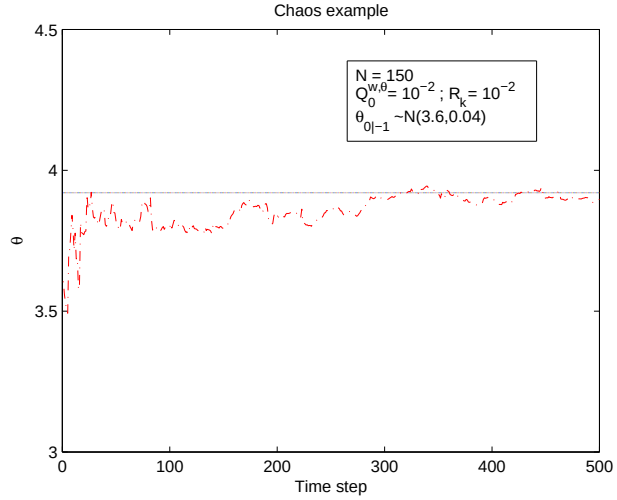

Figure 5.17: Estimate of θ with $Q_0^{w,\theta} = 10^{-2}$

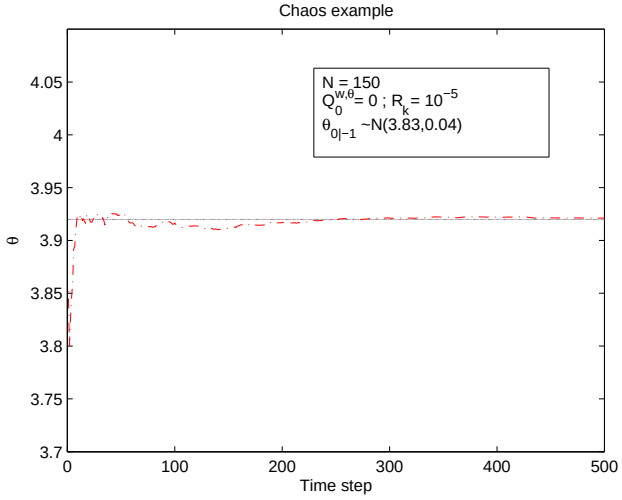

Figure 5.18: Estimate of θ with $Q_0^{w,\theta} = 0$

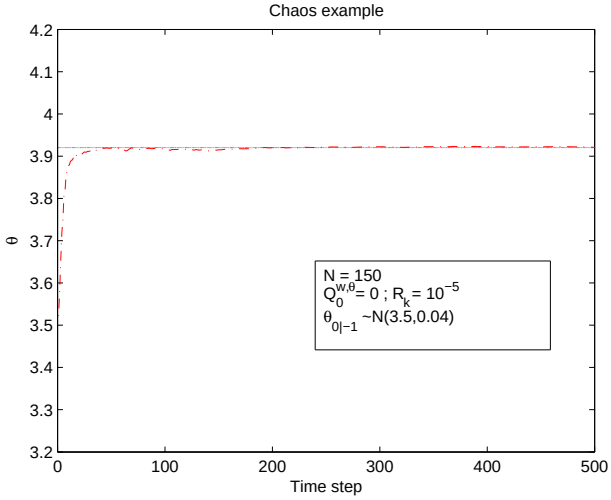

Figure 5.19: Estimate of θ with $Q_0^{w,\theta} = 0$

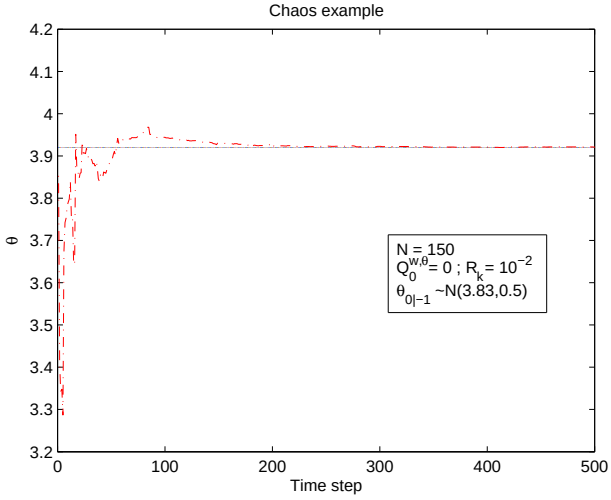

Figure 5.20: Estimate of θ with $Q_0^{w,\theta} = 0$

5.4.2 Time series example

We consider here the benchmark time series model with unknown parameter θ as given in equations (5.3.6)–(5.3.7). We note that this model can be written as

$$\begin{aligned} x_{k+1} &= f(x_k) + A(x_k)\theta_k + w_{k+1} && (5.4.6)\\ \theta_{k+1} &= \theta_k && (5.4.7)\\ y_k &= h(x_k) + C(x_k)\theta_k + v_k && (5.4.8) \end{aligned}$$

where $f(x_k) = \frac{x_k}{2} + 8\cos(1.2k)$, $A(x_k) = \frac{x_k}{1+x_k^2}$, $h(x_k) = \frac{x_k^2}{20}$ and $C(x_k) = 0$. In subsequent simulation study, we take $w_k \sim \mathcal{N}(0, 10)$, $v_k \sim \mathcal{N}(0, 1)$ and $p(x_0) \sim \mathcal{N}(0, 5)$. For the particle filter, we use 500 particles and the state transition density as proposal. The true parameter is $\theta^* = 25$. Starting with $\theta_{0|-1} \sim \mathcal{N}(10, 100)$, the estimate of θ, using marginalization method without roughening noise is shown in Figure 5.21. For comparison, we have also included the estimates obtained using roughening noise, η_k, in the parameter space. Like before, the variance of this roughening noise is reduced over each time step as described in Gustafsson and Hriljac (2003). For this example, we have considered two different cases with $\eta_0 \sim \mathcal{N}(0, 1)$ and $\eta_0 \sim \mathcal{N}(0, 0.1)$ and the corresponding estimates are shown in Figures 5.22–5.23 respectively. We observe that the parameter is converging in both the cases, though convergence seems to be slower than the case when there is no roughening noise.

In the next section, we consider the case where the number of available observations is limited and we describe another particle smoother based parameter estimation method, which is more efficient under such a scenario.

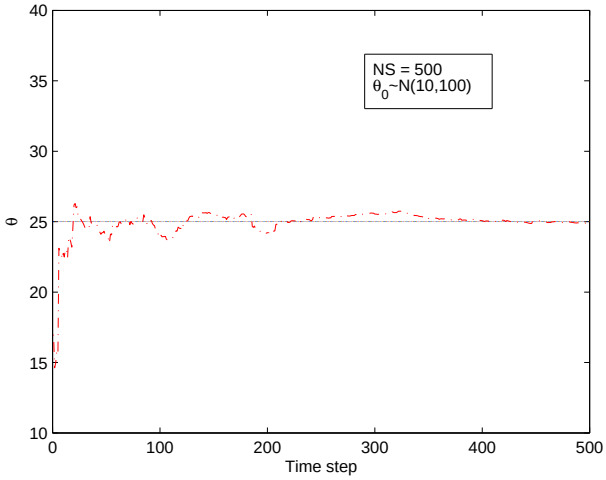

Figure 5.21: Estimate of θ with $\eta_k = 0$

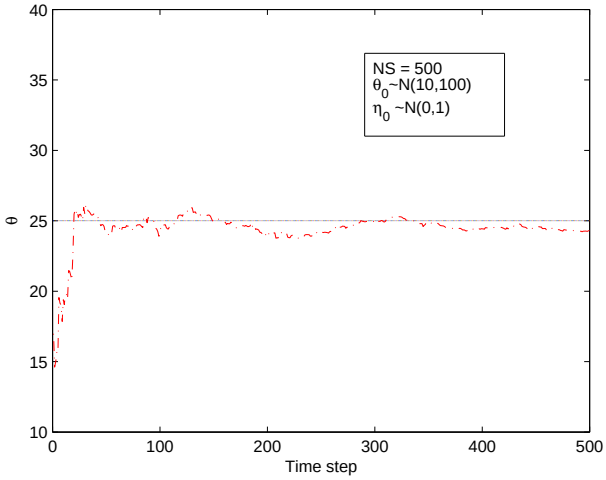

Figure 5.22: Estimate of θ with $\eta_0 \sim \mathcal{N}(0, 1)$

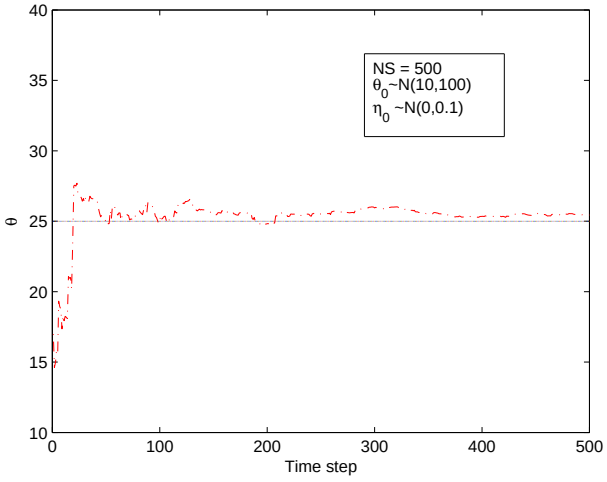

Figure 5.23: Estimate of θ with $\eta_0 \sim \mathcal{N}(0, 0.1)$

5.5 Parameter estimation using short observation data

Here we consider a different approach in dealing with the non dynamic nature of the unknown parameters. Furthermore, we assume that we have limited observation data. We cast the problem into a joint state estimation and model parameter identification framework. In our approach, rather than maximizing the likelihood of the observed data with respect to the parameters (as done in the ML estimation), we maximize the joint likelihood of the observation and unobserved state sequence with respect to both the unknown parameters and the unobserved state sequence. This criterion has been first considered by Bar-Shalom (1972) for linear-Gaussian case. See also Lim and Oppenheim (1978), Bagchi and ten Brummelhuis (1994) for similar approaches. However, the optimization steps for estimating the joint state sequence for general nonlinear and/or non Gaussian model is not trivial and as a result, a similar study involving a general state space model is missing in the literature. For the special cases, where closed form solutions for optimal state sequence can be obtained (for example, when both the state and observation sequences are jointly Gaussian), it is known that the estimate is biased, but for short observation data this method outperforms the ML estimate in terms of mean-squared error (MSE) (Yeredor (2000)). MSE is a direct measure of estimation error which takes both bias and variance into account and in many cases, biased estimates may result in an MSE that is smaller than the Cramer-Rao lower bound (CRLB), which characterizes the smallest achievable variance of any unbiased estimator. In fact, biased estimation methods are used extensively in different signal processing applications, specially with short/limited observation data and/or low signal to noise ratios (SNRs)(Kay and Eldar (2008)). Moreover, an unbiased estimator may not even exist in many cases or the unbiasedness requirement can lead to meaningless results. Thus, even though ML is an asymptotically efficient estimator (unbiased with minimum variance), for short data records, where ML is indeed incapable of achieving its asymptotic optimality, estimator based on minimum MSE may be preferable. To implement our proposed method for a general state space model, the crucial step is, as mentioned before, the estimation of the optimal joint state sequence, which is in general analytically intractable. We approximate this optimal state sequence using a particle based method developed in the recent past by Godsill *et al.* (2001). Thus, our contribution extends

the existing results for a linear-Gaussian case to a general state space problem. We give the formulation of our approach in the next section.

5.5.1 Problem formulation

Consider the discrete time state space model

$$x_t \sim p\left(\cdot|x_{t-1};\theta\right) \tag{5.5.1}$$

$$y_t \sim p\left(\cdot|x_t;\theta\right) \tag{5.5.2}$$

where at time t, x_t is the unobserved state and y_t is the observation. The static parameter vector $\theta = (\theta_1, \cdots, \theta_m) \in \Theta \subset \mathbb{R}^m$. $p\left(\cdot|\cdot\right)$ is a generic conditional probability density function (pdf). Now, given a relatively small set of observations $y_{1:T}$, our objective here is to extract information about the unknown θ. One conventional way to achieve this is the classical ML criterion, where one maximizes $l_T(\theta) \equiv p(y_{1:T};\theta)$ (also known as likelihood) with respect to θ to obtain $\widehat{\theta}_{ML}$, as

$$\arg\max_{\theta} p(y_{1:T};\theta) = \arg\max_{\theta} \int p(x_{0:T}, y_{1:T};\theta) dx_{0:T}. \tag{5.5.3}$$

In practice, however, the above marginalization step is often analytically intractable. On the other hand, the joint likelihood of the observations and unobserved state sequence (also known as complete likelihood) is easy to construct due to the Markovian nature of the model considered :

$$p(x_{0:T}, y_{1:T};\theta) = p(x_0;\theta)\prod_{t=1}^{T} p(x_t|x_{t-1};\theta)p(y_t|x_t;\theta). \tag{5.5.4}$$

Consequently, the complete log likelihood is then given by

$$\begin{aligned}\log(p(x_{0:T}, y_{1:T};\theta)) \;&=\; \log(p\left(x_0;\theta\right)) + \\ &+ \sum_{t=1}^{T} \log(p\left(x_t|x_{t-1};\theta\right)) + \sum_{t=1}^{T} \log(p\left(y_t|x_t;\theta\right)).\end{aligned} \tag{5.5.5}$$

In this section, we maximize the complete likelihood with respect to both the unknown parameter (θ) and the unobserved state sequence ($x_{0:T}$), rather than maximizing the likelihood of the observed data with respect to parameters. This leads to dividing the problem into two interconnected sub problems — subproblem A (state estimation) and subproblem B (parameter estimation). The details are described below.

5.5.1.1 Subproblem A

Estimation of (smoothed) state assuming the parameter values: First assume that $\theta = \theta^{old}$. The state estimation problem is then finding

$$\widehat{x}_{0:T} = \arg\max_{x_{0:T}} p(x_{0:T}, y_{1:T}; \theta^{old}). \tag{5.5.6}$$

Since

$$p(x_{0:T}|y_{1:T}; \theta) = \frac{p(x_{0:T}, y_{1:T}; \theta)}{p(y_{1:T}; \theta)} \tag{5.5.7}$$

and the denominator in equation (5.5.7) is independent of $x_{0:T}$, the state estimation problem in (5.5.6) can be cast into the usual maximum *a posteriori* (MAP) sequence of $x_{0:T}$ conditioned on observed $y_{1:T}$ and (assumed) θ^{old}. For a general nonlinear and/or non Gaussian state space model, this problem is nontrivial. However, one can approximate this MAP sequence using the particle based method developed by Godsill *et al.* (2001). The memory requirement of this MAP sequence algorithm is $O(NT)$ and the computational complexity is $O(N^2T)$. However, since we are dealing with short data regime, memory requirement or complexity is not a serious issue here.

5.5.1.2 Subproblem B

Estimation of the parameter assuming the state is known : Given all the observation data $y_{1:T}$ and the estimate of the state, $\widehat{x}_{0:T} \equiv \widehat{x}_{0:T}(y_{1:T}, \theta^{old})$ from sub problem A, one can obtain a new estimate of θ as

$$\theta^{new} = \widehat{\theta} = \arg\max_{\theta} p(\widehat{x}_{0:T}, y_{1:T}; \theta). \tag{5.5.8}$$

This maximization problem can be translated into finding the zeros of the gradient of the complete log likelihood. Define $L(\theta) \triangleq \log p(x_{0:T}, y_{1:T}; \theta)$. Then $\widehat{\theta}$ is a solution to

$$\nabla L(\theta) = 0 \tag{5.5.9}$$

where ∇ is the gradient vector (w.r.t θ). From (5.5.5) we get

$$\begin{aligned} \nabla L(\theta) &= \left[\frac{\nabla p\left(x_0;\theta\right)}{p\left(x_0;\theta\right)}\right]_{\substack{\widehat{x}_{0:T} \\ y_{1:T}}} + \sum_{t=1}^{T}\left[\frac{\nabla p\left(x_t|x_{t-1};\theta\right)}{p\left(x_t|x_{t-1};\theta\right)}\right]_{\substack{\widehat{x}_{0:T} \\ y_{1:T}}} \\ &\quad + \sum_{t=1}^{T}\left[\frac{\nabla p\left(y_t|x_t;\theta\right)}{p\left(y_t|x_t;\theta\right)}\right]_{\substack{\widehat{x}_{0:T} \\ y_{1:T}}}. \end{aligned} \tag{5.5.10}$$

Equation (5.5.9) and (5.5.10) lead to a system of nonlinear equations in θ and θ^{new} can be obtained by solving them. Typically $\Theta = \prod_{i=1}^{m}(\theta_{i,min}, \theta_{i,max})$ and if θ^{new} does not belong to Θ, to prevent divergence, a standard practice consists of reprojecting θ^{new} inside Θ. The reprojections are done such that if $\theta_i^{new} < \theta_{i,min}$, θ_i^{new} is set to $\theta_{i,min}$. Similarly, when $\theta_i^{new} > \theta_{i,max}$, θ_i^{new} is set to $\theta_{i,max}$ (Bar-Shalom (1972)).
Now, starting with an initial value $\theta^{(0)}$, the final solution is obtained by iterating between these two subproblems until a pre-specified stopping criterion is reached. The stopping criterion may be reached, for example, when either of the following is satisfied:

C1 : for a pre-determined Δ, when $\frac{\left\|\widehat{\theta}^{(k)} - \widehat{\theta}^{(k-1)}\right\|}{\left\|\widehat{\theta}^{(k-1)}\right\|} \leq \Delta$

C2 : when the iteration stage k reaches (pre-defined) maximum allowable stage (K_{Max}). This signifies that the iteration does not converge and one has to start afresh with new initial values of parameters.

This is essentially a batch method of estimating the parameters. However, estimates obtained through our method using limited observations can be used, for example, as initial values of the parameters in any online parameter estimation method. This (batch) method provides means to obtain parameter estimates without any addition of artificial process noise. Therefore, it can be expected that our method will provide more accurate parameter estimates than the augmented particle filter.

It is important to emphasize here the relation of the above algorithm to the popular EM algorithm (Wills *et al.* (2008); Schön *et al.* (2009); Ninness (2009); Kantas *et al.* (2009)). Although, the above algorithm appears to be very similar to the EM algorithm, however, as outlined in Yeredor (2000), the difference essentially lies in the parameter updating step. In the EM algorithm, to get the new update on the parameters, one maximizes the expectation of the log pdf. In other words, the parameters are updated as

$$\begin{aligned}\theta^{new} = \widehat{\theta} &= \arg\max_{\theta} \mathbb{E}_{\theta^{old}}[log\ p(x_{0:T}, y_{1:T}; \theta)] \\ &= \arg\max_{\theta} \int [log\ p(x_{0:T}, y_{1:T}; \theta)] p(x_{0:T}|y_{1:T}; \theta^{old}) dx_{0:T}.\end{aligned}$$

On the other hand, for our algorithm, the parameters are updated using the following iteration

$$\theta^{new} = \widehat{\theta} = \arg\max_{\theta} log\ p(\widehat{x}_{0:T}(\theta^{old}), y_{1:T}; \theta).$$

Thus, instead of maximizing the expectation of the log pdf, here we plug in the estimated MAP sequence $\widehat{x}_{0:T}(\theta^{old})$ and then maximize the same log pdf.

The EM is popular due to the fact that the sequence $\{l_T(\theta_k) \equiv p(y_{1:T};\theta_k)\}_k$ generated by the EM is guaranteed to be non decreasing i.e. $l_T(\theta_{k+1}) \geq l_T(\theta_k)$ (see for example, the article by Ninness (2009) for the proof). However, it is to be noted that if $\mathbb{E}_{\theta^{old}}[log\ p(x_{0:T}, y_{1:T};\theta)]$ is numerically approximated (which is often the case in the context of any particle filtering/smoothing based EM algorithm), this guarantee does not hold any longer (Kantas *et al.* (2009)). Moreover, the EM algorithm is essentially an approach for solving the maximum likelihood problem. Thus, when the available observation data is short, the EM algorithm has the same limitation as that of maximum likelihood method, i.e. the asymptotic optimal property does not hold.

In the following section, we illustrate our method by means of an example.

5.5.2 Simulation results

We demonstrate the proposed method on the benchmark time series model given by equations (5.2.6)–(5.2.7). Note that here, $\frac{\partial f(.)}{\partial \theta_1} = x_{k-1}$ and $\frac{\partial f(.)}{\partial \theta_2} = \frac{x_{k-1}}{1+x_{k-1}^2}$. Therefore, the use of (5.5.10) reduces (5.5.9) to

$$\begin{aligned}\sum_{k=1}^{T}\left[\{x_k - f(.)\}\frac{\partial f(.)}{\partial \theta_1}\right]_{\widehat{x}_{0:T}} &= 0\\ \sum_{k=1}^{T}\left[\{x_k - f(.)\}\frac{\partial f(.)}{\partial \theta_2}\right]_{\widehat{x}_{0:T}} &= 0.\end{aligned}$$

For simulations, true values are taken to be $\theta_1^* = 0.5$ and $\theta_2^* = 25$ respectively. We use $T = 50$, $N = 500$ particles and so called "Exact Moment Matching (EMM) proposal" (Saha *et al.* (2009*b*)). We start with $\theta_1^{(0)} = -0.3$ and $\theta_2^{(0)} = 10$. Here we perform 40 iterations regardless of stopping criteria and the estimated parameters with different iteration stages are shown in Figure 5.24 and Figure 5.25. If working with $\Delta = 0.005$, we would have stopped at iteration 7. We should note that both the estimates show distinct bias, even with increasing number of iteration. This is already observed in linear-Gaussian case (Yeredor (2000)). We further compare our approach to the augmented state space method where the

variance of the artificial noise is reduced over time as in Gustafsson and Hriljac (2003). For a qualitative comparison, we assume θ_2 fixed at its true value and estimate θ_1 when the number of available observation data is 50. Our estimate of θ_1 is similar to the previous example and we do not include it here. Here we note that for the augmented state method, the selection of initial variance of the artificial noise is not obvious and the convergence of the additional state (parameter) depends on this selection. Our method does not have this difficulty. For the subsequent implementation of the augmented method, we take $p(\theta_1^{(0)}) \sim$ uniform $(-0.5,\ 1.5)$, initial variance of the artificial noise $= 1$, $T = 1000$, $N = 2000$ and state transition density as proposal. As seen from Figure 5.26, estimate of θ_1 using augmented method, keeps oscillating even after large time steps. On the other hand, our method does converge quite rapidly, albeit with a distinct bias. If this bias can somehow be theoretically ascertained, then our method will decidedly give better result. Next, for a quantitative comparison, we consider the same dynamic system as defined in (5.2.6)–(5.2.7) and estimate $\theta \equiv (\theta_1, \theta_2)$. We then compare the estimates in terms of MSE over 100 Monte Carlo runs. For our approach, assuming $\Theta = (-1.5, 2.5) \times (15, 35)$, we start with $\theta_1^{(0)} = 0.1$ and $\theta_2^{(0)} = 15$, $N = 500$ particles, $T = 50$ and use EMM proposal. For each run, we perform 20 iterations irrespective of the stopping criteria. The MSE obtained is 65.5664. For the augmented method, the model is rewritten as

$$\begin{aligned} x_k &= \theta_{1,k-1} x_{k-1} + \frac{\theta_{2,k-1} x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2k) + w_k \\ \theta_{1,k} &= \theta_{1,k-1} + \eta_{1,k} \\ \theta_{2,k} &= \theta_{2,k-1} + \eta_{2,k} \\ y_k &= \frac{x_k^2}{20} + v_k, \end{aligned}$$

where $\eta_{1,k}$ and $\eta_{2,k}$ are the artificial noises, $k = 1, \cdots, T$. In vector notation, the above model can be expressed as

$$\begin{aligned} X_k &= g(X_{k-1}) + W_k \\ y_k &= h(X_k) + v_k, \end{aligned}$$

where $X_k = [x_k\ \theta_{1,k}\ \theta_{2,k}]^T$ and $W_k = [w_k\ \eta_{1,k}\ \eta_{2,k}]^T$ are new state and process noise vector respectively. We take $p(\theta_{1,0}) \sim$ uniform $(-1.5,\ 2.5)$, $p(\theta_{2,0}) \sim$ uniform $(15,\ 35)$ (as used for Θ above), $\eta_{1,k} \sim \mathcal{N}(0, \frac{0.1}{k})$, $\eta_{2,k} \sim$

$\mathcal{N}(0, \frac{5}{k})$, $N = 2000$, $T = 1000$ and state transition density as proposal. In 64 out of 100 runs, the algorithm gave meaningful results for the likelihood (i.e. $p(y_k|X_k)$). In the remaining situations, the likelihood became too small to contribute to the weights and the process stopped after a while. The problem perhaps lies with the fact that 2000 samples for X_k are not sufficient enough to explore the state space effectively. However, for this case, we took only those 64 runs for computing the MSE and the MSE obtained (over 64 runs) is 61.1643.
Although the above two methods are similar in terms of MSE as obtained here, we note that a direct comparison is not meaningful in the sense that we have discarded the remaining 36 runs for the augmented method. Also, the selection of initial variance of the artificial noises in the augmented method is not obvious and the convergence of the additional state (parameter) depends on this selection. On the other hand, our method can work with static parameter and as such, no artificial dynamics is required. Thus, despite the bias, our method is clearly preferable for short data series.

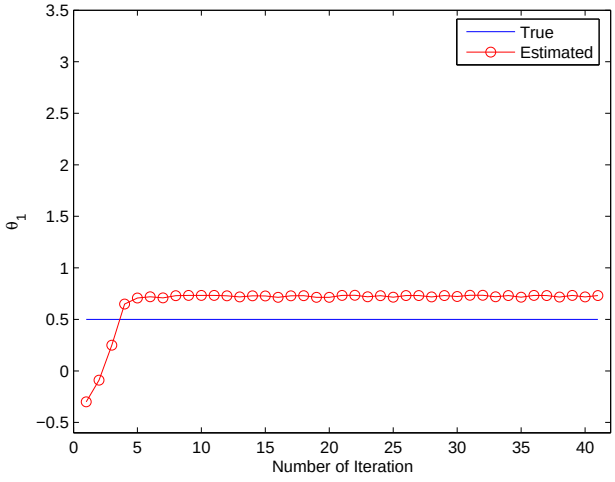

Figure 5.24: Estimate of θ_1 w.r.t. iteration stage.

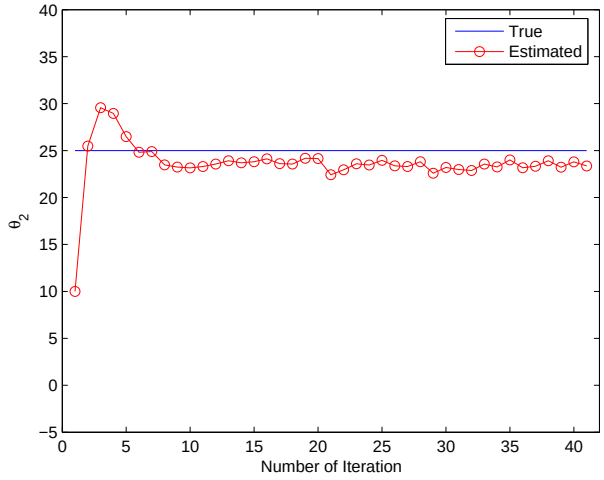

Figure 5.25: Estimate of θ_2 w.r.t. iteration stage.

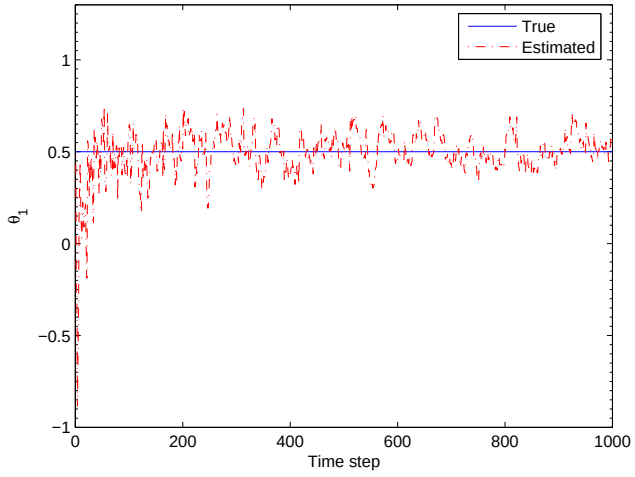

Figure 5.26: Estimate of θ_1 using augmented state method.

5.6 Concluding Remarks

In this chapter, we have introduced some new particle filtering/smoothing based schemes for estimating the unknown parameters of a general state space problem. We observe that these estimation procedures work quite well. We have also outlined the possible limitations of each method. Their performances are also illustrated through numerical simulations.

Chapter 6

Conclusions and Future Research

This chapter summarizes the main contributions of this thesis and presents some directions for further research.

6.1 Conclusions

In this thesis we have studied various aspects of particle filtering and smoothing algorithms. The particle filtering method is an elegant tool to approximate the posterior of some (unobserved) state process (x_k) from the noisy observation process (y_k) in the form of weighted random samples.

Since the particle filtering algorithm forms the basis of our subsequent works, a brief description of a generic particle filter algorithm is given in Chapter 2. It consists of three basic blocks – (a) generating the samples sequentially according to some importance function, (b) updating the weights of those samples given the recent observation and (c) resampling when necessary. The roles of importance sampling and resampling steps are also discussed.

Chapter 3 deals with the importance function. As the random samples representing the posterior are generated from this distribution, a proper selection of the importance function plays a key role for the efficiency of the underlying particle filtering method. Often the state transition density $p(x_k|x_{k-1})$, which is easily available from the model, is used as the importance function. However, with this choice, a lot of samples are wasted to

explore the state space, especially when the measurement is very informative. On the other hand, an importance function of the form $p(x_k|x_{k-1}, y_k)$ is a smarter choice as it also incorporates the recent observation. In fact, it is shown by Doucet *et al.* (2000) that this importance function is optimal in the sense that the conditional variance of the un-normalized importance weight is minimum. Later in chapter 5, we have seen that for some important problems in finance, this optimal importance function comes naturally due to the typical problem set up. In general, however, this is not the case and in practice, it is difficult to evaluate this function and/or to sample from it. We have introduced a new Gaussian importance function (EMM), which approximates the aforementioned optimal function using moment matching method. To be more specific, the EMM method is based on Gaussian approximation of the conditional distribution of (x_k, y_k), given x_{k-1}, with the first two moments matched exactly to those of the true conditional distribution. Thus, to use the proposed method, one needs to know the moments of the system dynamics up to the second order. This is satisfied, for example, when the noise processes are additive Gaussian and the observation equation is polynomial. When the exact moments are not known but the noise processes are additive Gaussian and the observation model is smooth, one can use a polynomial approximation of the observation model to derive this importance function. We have further compared our proposed EMM method with the other existing Gaussian importance functions (LIN, GHQ, JUQ, UPF), which also incorporate the latest observation in the importance function. The numerical results show that our EMM method provides better results when considering the trade off between the performance (RMSE) and the computational load. The performances of the different importance functions are also compared in terms of Kullback-Leibler divergence.

In Chapter 4, we have studied the problem of extracting the maximum *a posteriori* (MAP) estimate of the state using the weighted particle representation of the state posterior. The particle with the highest weight is often postulated as the MAP estimate (without proof or evidence) even in quite recent literature. However, it has been shown recently that this is not the true MAP estimator. Subsequently, a filter MAP estimator in the particle filtering set up has been introduced by Driessen and Boers (2008*a*). In this chapter, we have presented an estimator for the smoothed marginal MAP, which extracts the required MAP from the weighted particle cloud of the smoother. The particle cloud for the smoother can be generated

using either the forward-backward smoother or the generalized two filter smoother. We have presented algorithms for the smoothed marginal MAP using either of the smoothers. This smoothed marginal MAP estimator is further applied to estimate the unknown initial condition of a dynamic system. In this chapter, we have also explored a couple of possible improvements in the filter MAP algorithm proposed by Driessen and Boers (2008*a*). While Driessen and Boers (2008*a*) have maximized the filter posterior along the particles, we have studied the performance of gradient based optimization with the filter MAP of Driessen and Boers (2008*a*) as the starting point. We observe that the gradient based optimization makes the method unnecessarily compuationally intensive without providing much gain in terms of RMSE. On the other hand, by tweaking the number of particles in constructing the predictive density, the computational load can be highly reduced without affecting much on RMSE performance.

Chapter 5 addresses the issues of estimating the parameters of a general state space model. One very popular way of estimating the parameter in this framework, is to augment the parameter as an additional state driven by artificial noise. However, the introduction of these artificial dynamics turns the fixed parameter into a slowly varying one. In this chapter, we have proposed some particle filtering/smoothing schemes for estimating the parameter, which either avoids or minimizes the effect of the artificial dynamics within this augmented framework. Recall that the unknown initial condition of a state is estimated by the smoothed marginal MAP in Chapter 4. Furthermore, in the usual augmented framework, the initial augmented state (parameter) is not corrupted by any artificial noise. Subsequently, we have taken the smoothed marginal MAP of the initial augmented state to be the estimate of the true (fixed) parameter. Two other augmented methods, where no artificial noise is added, are also considered. The first one is based on the marginalized particle filter method, where the fact that the parameters enter in a linear way is exploited and subsequently a Kalman filter is used to estimate the parameters. For the other method, the parameter is augmented similarly without adding any artificial noise and particle filter is applied on the model. Furthermore, we have introduced another method, which is not an augmented method, to estimate the static parameter. This is based on maximizing the joint likelihood with respect to both the unknown parameter and the unknown state sequence. This leads to two interconnected subproblems – estimation of the

state given the parameter and then estimating the parameter conditioned on the state obtained from the previous subproblem. Thus, starting with an initial guess on the parameter, the final parameter estimate is obtained by iterating between these two subproblems until a convergence in parameter is obtained. Finally, we have considered a very challenging practical problem in mathematical finance, where we have estimated the unobserved volatility (along with the model parameter) using the celebrated stochastic volatility model of Heston including jumps. The model parameters are estimated using the augmented state space formulation without any parameter dynamics.

6.2 Future Research

In this section we review several issues which need urgent attention in the near future.

- In Chapter 3, we have proposed the importance function to be a Gaussian approximation of the optimal importance function $p(x_k|x_{k-1}, y_k)$ by matching the first two moments of the dynamical system. The advantage was that it is easily implementable (easy to sample from and evaluate). When the process noise is from a non-Gaussian family, which is still reasonably easy to implement as an importance function, one may take the importance function to be from the same family. In other words, by approximating the optimal importance function $p(x_k|x_{k-1}, y_k)$ by a distribution from the same family as that of $p(x_k|x_{k-1})$. In doing so, one needs to match as many moments as required to identify all the parameters of the distribution.
- The filter MAP for the time step k as described in Chapter 4 (pf-MAP), depends on the (weighted) particles cloud of the immediate past (i.e. on particle cloud at time step $k-1$). This is a constraint on the storage requirement. Therefore, one may consider developing a MAP estimator, other than using the kernel fitting method, which does not have this constraint.
- In the context of estimation of a fixed but unknown initial state (as treated in section 4.7.1.1), it would be interesting to study the effect of the assumed initial distribution (possibly dependent on unknown initial state) and further convergence issues.

- One may possibly extend the smoothed marginal MAP estimator as developed in Chapter 4, to a jump Markov system.
- In Chapter 5, we have applied the smoothed marginal MAP for estimating the parameter in the augmented framework. Here, we have followed the usual step of augmenting the parameter as additional state driven by artificial noise. However, the selection of the variance of this artificial noise is not obvious. It would be interesting to study the sensitivity of the estimate of the parameter on this selection.
- In the parameter estimation problem with large number of parameters, if one uses the augmented method, it is very difficult to explore the state space effectively with limited number of particles. This can be ascribed to the fact that the additional states (parameters) are independent of each other and the observation carries the information regarding these additional states indirectly via the true state. As a result, it becomes very difficult to explore the region (in this high dimensional space) where the joint posterior density would be high. Therefore further research attention is needed to address this problem.
- The whole issue of convergence analysis is a major research area where more research effort is needed.

Appendix A

Derivation of Importance function by linearization (LIN)

Let us consider a system dynamics with linear observation equation and additive Gaussian noises :

$$\begin{aligned} x_k &= f(x_{k-1}) + w_k, \qquad w_k \sim \mathcal{N}(0, Q) \\ y_k &= c_1 + c_2 x_k + v_k, \qquad v_k \sim \mathcal{N}(0, R). \end{aligned}$$

Further assume that w_k and v_k are independent. We shall show that the conditional distribution of x_k given x_{k-1} and y_k is Gaussian with appropriate mean and variance.

To that end, we first derive the joint distribution of (x_k, y_k) conditional upon x_{k-1}. Note that y_k can be rewritten as

$$y_k = c_1 + c_2 f(x_{k-1}) + c_2 w_k + v_k,$$

so that

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} f(x_{k-1}) \\ c_1 + c_2 f(x_{k-1}) \end{pmatrix} + \begin{pmatrix} I & 0 \\ c_2 & I \end{pmatrix} \cdot \begin{pmatrix} w_k \\ v_k \end{pmatrix}.$$

We rewrite this as

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = A + B \begin{pmatrix} w_k \\ v_k \end{pmatrix},$$

where A is a vector and B is a matrix given by

$$A = \begin{pmatrix} f(x_{k-1}) \\ c_1 + c_2 f(x_{k-1}) \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} I & 0 \\ c_2 & I \end{pmatrix}.$$

Hence from the theory of multivariate Gaussian distribution it follows that given x_{k-1} the conditional distribution of

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right),$$

where

$$\mu_1 = f\left(x_{k-1}\right), \qquad \mu_2 = c_1 + c_2 f\left(x_{k-1}\right), \tag{A.0.1}$$

$$\Sigma_{11} = Q, \qquad \Sigma_{12} = Q c_2^T = \Sigma_{21}^T, \qquad \Sigma_{22} = c_2 Q c_2^T + R. \tag{A.0.2}$$

Furthermore, $f(x_k|x_{k-1}, y_k) \sim \mathcal{N}(m_k, V_k)$, with

$$m_k = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_k - \mu_2), \quad \text{and} \tag{A.0.3}$$

$$V_k = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \tag{A.0.4}$$

Now, using the following matrix identity (Bagchi (1993), equations (3.12a)-(3.12c), page 60)

$$(A - BD^{-1}B^T)^{-1} = A^{-1} + A^{-1}B\left(D - B^T A^{-1} B\right)^{-1} B^T A^{-1}$$

we see

$$\begin{aligned} V_k^{-1} &= \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1} \\ &= \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^T\right)^{-1} \\ &= \Sigma_{11}^{-1} + \Sigma_{11}^{-1}\Sigma_{12}\left(\Sigma_{22} - \Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12}\right)^{-1}\Sigma_{12}^T\Sigma_{11}^{-1} \\ &= Q^{-1} + Q^{-1}Qc_2^T\left(c_2 Q c_2^T + R - c_2 Q Q^{-1} Q c_2^T\right)^{-1} c_2 Q Q^{-1} \\ &= Q^{-1} + c_2^T R^{-1} c_2. \end{aligned} \tag{A.0.5}$$

Also, from (A.0.3), using (A.0.1), (A.0.2), (A.0.4), and (A.0.5) we have

$$\begin{aligned}
m_k &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_k - \mu_2) \\
&= \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}c_2 f\left(x_{k-1}\right) + \Sigma_{12}\Sigma_{22}^{-1}(y_k - c_1) \\
&= f\left(x_{k-1}\right) - \Sigma_{12}\Sigma_{22}^{-1}c_2 f\left(x_{k-1}\right) + Qc_2^T\Sigma_{22}^{-1}(y_k - c_1) \\
&= \left(I - \Sigma_{12}\Sigma_{22}^{-1}c_2\right) f\left(x_{k-1}\right) + V_k V_k^{-1} Q c_2^T \Sigma_{22}^{-1}(y_k - c_1) \\
&= \left(I - \Sigma_{12}\Sigma_{22}^{-1}c_2\right) QQ^{-1} f\left(x_{k-1}\right) + \\
&\qquad + V_k\left(Q^{-1} + c_2^T R^{-1} c_2\right) Q c_2^T \Sigma_{22}^{-1}(y_k - c_1) \\
&= \left(Q - \Sigma_{12}\Sigma_{22}^{-1}c_2 Q\right) Q^{-1} f\left(x_{k-1}\right) + \\
&\qquad + V_k\left(c_2^T + c_2^T R^{-1} c_2 Q c_2^T\right)\Sigma_{22}^{-1}(y_k - c_1) \\
&= \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right) Q^{-1} f\left(x_{k-1}\right) + \\
&\qquad + V_k c_2^T R^{-1}\left(R + c_2 Q c_2^T\right)\Sigma_{22}^{-1}(y_k - c_1) \\
&= V_k Q^{-1} f\left(x_{k-1}\right) + V_k c_2^T R^{-1}\Sigma_{22}\Sigma_{22}^{-1}(y_k - c_1) \\
&= V_k Q^{-1} f\left(x_{k-1}\right) + V_k c_2^T R^{-1}(y_k - c_1) \\
&= V_k\left[Q^{-1} f\left(x_{k-1}\right) + c_2^T R^{-1}(y_k - c_1)\right] \qquad \text{(A.0.6)}
\end{aligned}$$

Now, to obtain the expressions (3.4.2) and (3.4.3) of section 3.4.1, recall, from (3.4.1), that the linearized observation equation is given by

$$y_k \approx h(f(x_{k-1})) + C_k(x_k - f(x_{k-1})) + v_k,$$

so that

$$c_1 = h(f\left(x_{k-1}\right)) - C_k f\left(x_{k-1}\right) \text{ and } c_2 = C_k.$$

Appendix B

Particle filtering implementation of the Bates model with optimal importance function

Here we describe the so called Bates model for volatility in mathematical finance and work out the details regarding the implementation of the particle filter. The model can be described as follows. The (observed) stock price (S_t) evolves according to:

$$dS_t = \mu_S S_t dt + \sqrt{v_t} S_t dB_t + S_t dZ_t^J - \lambda m^J S_t dt, \quad \text{(B.0.1)}$$

$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dZ_t \quad \text{(B.0.2)}$$

where v_t is the unobserved volatility, B_t and Z_t are standard Brownian motion processes with correlation ρ. Z_t^J denotes the pure-jump process which contains two components: random jump-event times and random jump sizes. The jump-event times $\{T_i; i \geq 1\}$ arrive with a constant intensity λ . Given the arrival of the i-th jump event, the stock price jumps from $S_{T_i^-}$ to $S_{T_i^-} \exp(U_i^s)$ where U_i^s is normally distributed with mean μ_J and variance σ_J^2, independent of B_t and Z_t, inter-jump times and of U_j^s, for $j \neq i$. Intuitively, the conditional probability at time t of another jump before $t + \Delta t$ is, for some small Δt, approximately $\lambda \Delta t$ and, conditional on a jump event, the mean relative jump size is $m^J = E(\exp(U^s) - 1) = \exp(\mu_J + \sigma_J^2/2) - 1$. Combining the effects of

random jump timing and size, the last term $\lambda m^J S_t dt$ in (B.0.1) compensates for the instantaneous change in expected stock introduced by the pure-jump process Z_t^J.

It follows that the logarithm of the stock $y_t = \log(S_t/S_0)$ satisfies

$$dy_t = (\mu_S - \lambda m^J - \frac{1}{2}v_t)dt + \sqrt{v_t}dB_t + dq_t^J, \tag{B.0.3}$$

where q_t^J is a compound Poisson process with intensity λ and the jump size is normally distributed as $\mathcal{N}(\mu_J, \sigma_J^2)$. Our aim is to estimate recursively the volatility process v_t and the unknown system parameters from the observed data $\{y_{s;\ 0\leq s\leq t}\}$. By introducing another standard Brownian motion process $\tilde{Z}_t$, which is independent of B_t, we can express dZ_t as

$$dZ_t = \sqrt{1-\rho^2}d\tilde{Z}_t + \rho dB_t. \tag{B.0.4}$$

Now using dB_t as obtained from equation (B.0.3), one can write

$$dZ_t = \sqrt{1-\rho^2}d\tilde{Z}_t + \frac{\rho}{\sqrt{v_t}}(dy_t \tag{B.0.5}$$

$$-(\mu_S - \lambda m^J - \frac{1}{2}v_t)dt - dq_t^J). \tag{B.0.6}$$

Equation (B.0.2) can now be expressed as

$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t}\sqrt{1-\rho^2}d\tilde{Z}_t + \rho\xi(dy_t - (\mu_S - \lambda m^J - \frac{1}{2}v_t)dt - dq_t^J). \tag{B.0.7}$$

In order to apply the particle filtering algorithm to the (continuous time) state space model given by (B.0.3) and (B.0.7), we need to discretize the system first. We do this using Euler scheme. We select this scheme mainly due to its relative simplicity and limited computational load. However, while the continuous process itself is always non negative, the discretization is not (Lord *et al.* (2006)). As a result, precautionary care has to be taken to ensure the non negativity of the (discretized) state evolution. It is to be noted here that for implementing a particle filter, one needs to specify only the probabilistic description of state transition (together with initial state) and observation models. In this case, the exact state transition density is known analytically, which is non central chi

square and it is possible to simulate exactly from such a process (Broadie and Kaya (2006)). However, this is computationally involved and we chose to work only with the discretized version of the model. The discretization is done as follows:

$$\begin{aligned} y_k - y_{k-1} &= (\mu_S - \lambda m^J - \tfrac{1}{2}v_k)\Delta t \\ &\quad + \sqrt{v_{k-1}}\Delta B_k + \Delta q_k^J, \end{aligned} \tag{B.0.8}$$

where Δq_k^J is the jump in $q_{t_k}^J$. Furthermore,

$$\begin{aligned} v_k - v_{k-1} &= \kappa(\theta - v_{k-1})\Delta t + \xi\sqrt{v_{k-1}}\sqrt{1-\rho^2}\Delta\tilde{Z}_k \\ &\quad + \xi\rho(y_k - y_{k-1} - (\mu_S - \lambda m^J - \frac{1}{2}v_{k-1})\Delta t - \Delta q_k^J). \end{aligned} \tag{B.0.9}$$

We observe here that the (discrete time) state space model given by (B.0.8)–(B.0.9) does not have the Markovian structure that is commonly assumed. We, therefore, derive here explicitly the optimal importance function and the weight update equation.

Note that if we work with sequential importance sampling i.e. equation (2.3.13) holds, then from (2.3.6) and (2.2.5), it follows that

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|v_{0:k}^{(i)}, y_{0:k-1})p(v_k^{(i)}|v_{0:k-1}^{(i)}, y_{0:k-1})}{\pi(v_k^{(i)}|v_{0:k-1}^{(i)}, y_{0:k})}. \tag{B.0.10}$$

The optimal importance function in the sense that the variance of the weights conditioned on simulated past trajectory $v_{0:k-1}$ and $y_{0:k}$ is minimum (Doucet *et al.* (2000)), can be obtained here as

$$\pi(v_k|v_{0:k-1}, y_{0:k}) = p(v_k|v_{k-1}, y_{k-1:k}). \tag{B.0.11}$$

Assuming that at any discretization time period, the number of jump occurring is at most one, we have from Cont and Tankov (2004),

$$\begin{aligned} p(v_k|v_{k-1}, y_{k-1:k}) &= \{\frac{(1 - e^{-\lambda\Delta t}\lambda\Delta t)}{\sqrt{2\pi\bar{\sigma}^2(v_{k-1})}} \\ &\quad \times \exp[-\frac{(v_k - \bar{m}(v_{k-1}, y_{k-1:k}))^2}{2\bar{\sigma}^2(v_{k-1})}] \\ &\quad + \frac{e^{-\lambda\Delta t}\lambda\Delta t}{\sqrt{2\pi(\bar{\sigma}^2(v_{k-1}) + \xi^2\rho^2\sigma_J^2)}} \\ &\quad \times \exp[-\frac{(v_k - \bar{m}(v_{k-1}, y_{k-1:k}) + \xi\rho\mu_J)^2}{2(\bar{\sigma}^2(v_{k-1}) + \xi^2\rho^2\sigma_J^2)}]\}, \end{aligned} \tag{B.0.12}$$

where

$$\begin{aligned}\bar{m}(v_{k-1}, y_{k-1:k}) &= v_{k-1} + \kappa(\theta - v_{k-1})\Delta t \\ &\quad + \rho\xi(y_k - y_{k-1} - (\mu_S - \lambda m^J - \frac{1}{2}v_{k-1})\Delta t)\end{aligned}$$

and

$$\bar{\sigma}^2(v_{k-1}) = \xi^2 v_{k-1}(1-\rho^2)\Delta t.$$

Hence we obtain the optimal importance function for the particle filter. Additionally, in order to ensure that the volatility v_k is non negative, we enforce *a priori* constraint such that the samples selected from this proposal are all positive.

To complete the weight update equation (B.0.10), we need to calculate $p(v_k|v_{0:k-1}, y_{0:k-1})$ and $p(y_k|v_{0:k}, y_{0:k-1})$. Substituting $(y_k - y_{k-1})$ from equation (B.0.8) into equation (B.0.9), we get

$$\begin{aligned}v_k &= (1 + \frac{1}{2}\xi\rho\Delta t)^{-1}\{v_{k-1} + \kappa(\theta - v_{k-1})\Delta t \\ &\quad + \frac{1}{2}\xi\rho v_{k-1}\Delta t + \xi\sqrt{v_{k-1}}\sqrt{1-\rho^2}\Delta\tilde{Z}_k \\ &\quad + \xi\rho\sqrt{v_{k-1}}\Delta B_k\}.\end{aligned}$$

Hence, $p(v_k|v_{0:k-1}, y_{0:k-1}) = p(v_k|v_{k-1})$. This is expected as v_k does not contain any jump. Consequently, we have

$$p(v_k|v_{k-1}) = \mathcal{N}(\tilde{m}(v_{k-1}), \tilde{\sigma}(v_{k-1}))$$

where

$$\begin{aligned}\tilde{m}(v_{k-1}) &= (1 + \frac{1}{2}\xi\rho\Delta t)^{-1}\{v_{k-1} + \kappa(\theta - v_{k-1})\Delta t \\ &\quad + \frac{1}{2}\xi\rho v_{k-1}\Delta t\}\end{aligned}$$

and

$$\tilde{\sigma}(v_{k-1}) = (1 + \frac{1}{2}\xi\rho\Delta t)^{-1}\xi\sqrt{v_{k-1}}\sqrt{\Delta t}.$$

For the observation likelihood function $p(y_k|v_{0:k}, y_{0:k-1})$, we note from (B.0.8) that

$$p(y_k|v_{0:k}, y_{0:k-1}) = p(y_k|y_{k-1}, v_k, v_{k-1}).$$

which can be obtained as follows:

$$p(y_k|y_{k-1}, v_{k-1:k}) = \frac{(1 - e^{-\lambda\Delta t}\lambda\Delta t)}{\sqrt{2\pi v_{k-1}\Delta t}} \times \exp(-\frac{(y_k - (y_{k-1} + (\mu_S - \lambda m^J - \frac{1}{2}v_k)\Delta t + u_k))^2}{2v_{k-1}\Delta t}) + \frac{e^{-\lambda\Delta t}\lambda\Delta t}{\sqrt{2\pi(v_{k-1}\Delta t + \sigma_J^2)}} \times \exp(-\frac{(y_k - (y_{k-1} + (\mu_S - \lambda m^J - \frac{1}{2}v_k)\Delta t + u_k) - \mu_J)^2}{2(v_{k-1}\Delta t + \sigma_J^2)}).$$

where

$$u_k = \frac{\rho}{\xi}[v_k - v_{k-1} - \kappa(\theta - v_{k-1})\Delta t].$$

Bibliography

Aihara, S. I., A. Bagchi and S. Saha (2008). Estimating volatility and model parameters of stochastic volatility models with jumps using particle filter. In: *Proceedings of 17th IFAC World Congress*. Seoul, South Korea.

Aihara, S. I., A. Bagchi and S. Saha (2009). On parameter estimation of stochastic volatility models from stock data using particle filter - application to AEX index. *International Journal of Innovative Computing, Information and Control* **5**(1), 17–27.

Akashi, H. and H. Kumamoto (1975). Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques. *Systems and Control* **19**(4), 211–221. (In Japanese).

Anderson, B. D. O. and J. B. Moore (1979). *Optimal Filtering*. Prentice Hall. Englewood Cliffs, N.J.

Andrieu, C., A. Doucet and V. B. Tadić (2005). Online simulation-based methods for parameter estimation in nonlinear non Gaussian state space models. In: *Proceedings of IEEE Conference on Decision and Control*.

Andrieu, C. and A. Doucet (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society Series B* **64**(4), 827–836.

Arulampalam, S., S. Maskell, N. Gordon and T. Clapp (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing* **50**(2), 174–188.

Bagchi, A. (1993). *Optimal Control of Stochastic Systems*. Series in Systems and Control Engineering. Prentice Hall International.

Bagchi, A. and P. ten Brummelhuis (1994). Parameter identification in tidal models with uncertain boundaries. *Automatica* **30**(5), 745–759.

Bar-Shalom, Y. (1972). Optimal simultaneous state estimation and parameter identification in linear discrete time systems. *IEEE Transaction on Automatic Control* **17**(3), 308–319.

Bar-Shalom, Y. and X. Li (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. Academic press. New York.

Bates, D. (1996). Jumps and stochastic volatility: the exchange rate processes implicit in deutschemark option. *The Review of Financial Studies* **9**(1), 69–107.

Blom, H. A. P., E. A. Bloem, Y. Boers and J. N. Driessen (2008). Tracking closely spaced targets:Bayes outperformed by an approximation?. In: *Proc. of the 11th International Conference on Information Fusion*. Cologne, Germany.

Boers, Y. and J. N. Driessen (2003). A Particle filter based detection schemes. *IEEE Signal Processing Letters* **10**(10), 300–302.

Boers, Y. and J. N. Driessen (2007). The Mixed Labelling Problem in Multi- target Particle Filtering. In: *Proc. of the 10th International Conference on Information Fusion*. Quebec, Canada.

Boers, Y., E. Sviestins and J. N. Driessen (2008*a*). Mixed Labelling in Multi Target Particle Filtering. *IEEE Transactions on Aersospace and Electronic Systems*. Accepted for publication.

Boers, Y., J. N. Driessen and A. Bagchi (2009). Particle filter based MAP estimation for jump Markov systems. Working Paper, Draft version available on : wwwhome.math.utwente.nl/ boersy/mmap.pdf.

Boers, Y., J. N. Driessen and L. Schipper (2008*b*). Particle Filter Based Sensor Selection in Binary Sensor Networks. In: *Proc. of the 11th International Conference on Information Fusion*. Cologne, Germany.

Briers, M., A. Doucet and S. Maskell (2004). Smoothing algorithm for state space models. *Tech. Report CUED/F-INFENG/TR.498*.

Broadie, M. and Ö. Kaya (2006). Exact Simulation of Stochastic Volatility and other Affine Jump Diffusion Processes. *Operations Research* **54**(2), 217–231.

Candy, J. V. (2007). Bootstrap particle filtering. *IEEE Signal Processing Magazine* **24 (4)**, 73–85.

Cappé, O., S. J. Godsill and E. Moulines (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings* **95 (5)**, 899–924.

Carlin, B. P., M. G. Polson and D. S. Stoffer (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modelling. *Journal of the American Statistical Association* **87**(418), 493–500.

Chen, R. and J. Liu (2000). Mixture Kalman filters. *Journal of the Royal Statistical. Society Series B* **62**(3), 493–508.

Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada, available at http://soma.crl.mcmaster.ca/ zhechen/download/ieeebayesian.ps.

Cont, R. and P. Tankov (2004). *Financial Modelling With Jump Processes*. Chapman and Hall/CRC, Boca Raton.

Cover, T. M. and J. A. Thomas (1991). *Elements of Information Theory*. John wiley and Sons Inc.

Crisan, D. and A. Doucet (2002). A survey of convergence results on particle filtering for practitioners. *IEEE Transaction on Signal Processing* **50**(3), 736–746.

Del Moral, P. (2004). *Feynman-Kac Formulae : Genealogical and Interacting Particle Systems with Applications*. Springer Verlog New York.

Djurić, P. M. and J. Miguez (2002). Sequential particle filtering in the presence of additive gaussian noise with unknown parameters. In: *Proc. of the IEEE ICASSP*. pp. 803–806.

Doucet, A. and A. M. Johansen (2009). *A Tutorial on Particle Filtering and Smoothing: Fifteen years Later*. Oxford University Press. To appear in Oxford Handbook of Nonlinear Filtering.

Doucet, A. and V. B. Tadić (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics* **55**(2), 409–422.

Doucet, A., de Freitas, J.F.G. and Gordon, N.J., Eds. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag. New York.

Doucet, A., S. Godsill and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* **10**, 197–208.

Driessen, J. N. and Y. Boers (2008*a*). Particle filter MAP estimation in dynamical systems. In: *The IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*. Birmingham, UK.

Driessen, J. N. and Y. Boers (2008*b*). MAP Estimation in Nonlinear Dynamic Systems. Submitted to IEEE Transactions on Signal Processing.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* **57**(6), 1317–1339.

Gilks, W. R. and C. Berzuini (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B* **63**, 127–146.

Giunta, A. A., R. Narducci, S. Burgee, B. Grossman, W.H. Mason, L.T. Watson and R.T. Haftka (1995). Variable Complexity Response Surface Aerodynamic Design of an HSCT Wing. In: *Proceeding of the 13th AIAA Applied Aerodynamics Conference, San Diego, CA*. pp. 994–1002.

Godsill, S., A. Doucet and M. West (2001). Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics* **53**, 82–96.

Golub, G. H. (1973). Some modified matrix eigenvalue problems. *SIAM Review* **15**(2, Part 1), 318–334.

Gordon, N. J., D. J. Salmond and A. F. M. Smith (1993). Novel approach to nonlinear/ non Gaussian Bayesian state estimation. *IEE Proc. F-Radar and Signal Processing* **140**(2), 107–113.

Gray, R. M. (1990). *Entropy and Information Theory*. Springer Verlag, New York.

Guo, D., X. Wang and R. Chen (2005). New sequential Monte Carlo methods for nonlinear dynamic systems. *Statistics and Computing* **15**(2), 135–147.

Gustafsson, F. (2009). Particle filter Theory and Practice. *IEEE Transactions on Aerospace and Electronic Systems*. Accepted.

Gustafsson, F. and P. Hriljac (2003). Particle filters for system identification with application to chaos prediction. In: *Proceedings of SYSID03*. Rotterdam, The Netherlands.

Handschin, J. E. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica* **6**, 555–563.

Handschin, J. E. and D. Q. Mayne (1969). Monte Carlo techniques to estimate the conditional expectation in multistage nonlinear filtering. *Int. J. Control* **9**(5), 547–559.

Hol, J. D., T. B. Schön and F. Gustafsson (2006). On resampling algorithms for particle filters. In: *Proceedings of NSSPW 2006, Cambridge*. IEEE.

Hu, X., T. Schön and L. Ljung (2008). A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing* **56**(4), 1337–1348.

Hürzeler, M. and H. R. Künsch (1998). Monte Carlo Approximations for General State-Space Models. *Journal of Computational and Graphical Statistics* **7 (2)**, 175–193.

Ihler, A. (2005). Kernel density estimation toolbox for matlab (R13). available at http://www.ics.uci.edu/ ihler. Publicly available Matlab code.

Isard, M. and A. Blake (1998). A smoothing filter for condensation. In: *Proc. of 5th European Conference on Computer Vision*. pp. 767–781.

Ito, K. and K. Xiong (2000). Gaussian Filters for Nonlinear Filtering Problems. *IEEE Transaction on Automatic Control* **45**(5), 910–927.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press. New York.

Julier, S. J. and J. K. Uhlmann (1997). A new extensions of the Kalman filter to nonlinear systems. In: *Proceedingss of AeroSense :The 11th International Symposium on Aerospace/Defense Sensing,Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*. SPIE. pp. 182–193.

Julier, S. J., J. K. Uhlmann and H. F. Durrant-Whyte (2000). A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. *IEEE Transaction on Automatic Control* **45**(3), 477–482.

Kalnins, K., G. Jekabsons, J. Janushevskis, J. Auzins and R. Rikards (2006). Different polynomial functions in surrogate modelling of sandwich structures. In: *2nd International Workshop on surrogate Modelling and Space Mapping For Engineering Optimization (SMSMEO-06)*.

Kantas, N., A. Doucet, S. S. Singh and J. M. Maciejowski (2009). An Overview of Sequential Monte Carlo Methods forParameter Estimation in General State-Space Models. In: *Proceedings of SYSID09*. IFAC. France.

Karlsson, R., T. Schön and F. Gustafsson (2005). Complexity analysis of the marginalized particle filter. *IEEE Transaction on Signal Processing* **53**(11), 4408–4411.

Kay, S. and Y. C. Eldar (2008). Rethinking Biased Estimation. *IEEE Signal Processing Magazine* **25**(3), 133–136.

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association* **82**(400), 1032–1063.

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

Klaas, M., M. Briers, N. de Freitas, S. Maskell and D. Lang (2006). Fast Particle Smoothing: if I had a million particles. In: *Proc. of the 23rd International Conference on Machine Learning*. Pittsburgh, Pennsylvania. pp. 481–488.

Koehler, J. R. and A. B. Owen (1996). Computer Experiments. *Handbook on Statistics* **13**, 239–245.

Kong, A., J. S. Liu and W. H. Wong (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association* **89**(425), 278–288.

Lim, J. S. and A. V. Oppenheim (1978). All-pole modeling of degraded speech. *IEEE Transaction on Acoustics, Speech and Signal Processing* **26**, 197–210.

Liu, J. and M. West (2001). *Combined parameter and state estimation in simulation-based filtering*. Sequential Monte Carlo Methods in Practice. Springer-Verlag. New York.

Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* **93**(443), 1032–1044.

Lord, R., R. Koekkoek and D. van Dijk (2006). A comparison of biased simulation schemes for stochastic volatility models. Working paper.

Musso, C., N. Oudjane and F. Legland (2001). *Improving regularized particle filters*. Sequential Monte Carlo Methods in Practice. Springer New York.

Ninness, B. (2009). Some system identification challenges and approaches. In: *Proceedings of SYSID 2009*. IFAC. Saint-Malo, France.

Nordlund, Per-Johan (2008). Efficient Estimation and Detection Methods for Airborne Applications. PhD thesis. Department of Electrical Engineering, Linköping University.

Olsson, J. and T. Rydén (2008). Asymptotic properties of particle filter-based maximum likelihood estimators for state space models. *Stochastic Processes and their Applications* **118**(4), 649–680.

Pitt, M. K. and N. Shephard (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association* **94**(446), 590–599.

Ristic, B., S. Arulampalam and N. Gordon (2004). *Beyond the Kalman Filter : Particle Filter for Tracking Applications*. Artech House.

Saha, S., P. K. Mandal, A. Bagchi, Y. Boers and H. Driessen (2008*a*). On the Monte Carlo Marginal MAP Estimator for General State Space Models. Technical Report Memorandum 1887,. Department of Applied Mathematics, University of Twente. Enschede. ISSN 1874-4850.

Saha, S., P. K. Mandal, A. Bagchi, Y. Boers and H. Driessen (2009*a*). Parameter estimation in a general state space model from short observation data: A SMC based approach. In: *IEEE workshop on Statistical Signal Processing (SSP), Accepted*.

Saha, S., P. K. Mandal and A. Bagchi (2008*b*). A new approach to particle based smoothed marginal MAP. In: *Proceedings of EUSIPCO 2008*.

Saha, S., P. K. Mandal, Y. Boers and H. Driessen (2006). Exact moment matching for efficient importance functions in SMC methods. In: *Proceedings of NSSPW 2006, Cambridge*.

Saha, S., P. K. Mandal, Y. Boers, H. Driessen and A. Bagchi (2007). Gaussian proposal density using moment matching in SMC methods. Technical Report Memorandum 1832,. Department of Applied Mathematics, University of Twente. Enschede. ISSN 1874-4850.

Saha, S., P. K. Mandal, Y. Boers, H. Driessen and A. Bagchi (2009*b*). Gaussian proposal density using moment matching in SMC methods. *Statistics and Computing* **19**(2), 203–208.

Saha, S., Y. Boers, H. Driessen, P. K. Mandal and A. Bagchi (2009*c*). Particle based MAP state estimation: A comparison. In: *12th International Conference on Information Fusion (FUSION), Accepted.*

Schön, T., A. Wills and B. Ninness (2009). System identification of nonlinear state space models. Submitted to Automatica.

Schön, T. and F. Gustafsson (2003). Particle filters for system identification of state-space models linear in either parameters or states. Proceedings of SYSID 2003.

Schön, T., F. Gustafsson and P. J. Nordlund (2005). Marginalized particle filter for mixed linear/nonlinear state space models. *IEEE Transaction on Signal Processing* **53**(7), 2279–2289.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC.

Storvik, G. (2002). Particle Filters for State-Space Models With the Presence of Unknown Static Parameters. *IEEE Transaction on Signal Processing* **50**(2), 281–289.

van der Merwe, R., N. de Freitas, A. Doucet and E. Wan (2000). The unscented particle filter. Technical Report CUED/F-INFENG/TR 380. Cambridge University Engineering Department, Cambridge.

Wan, E. and R. van der Merwe (2000). The Unscented Kalman filter for nonlinear estimation. In: *The proceedings of IEEE symposium 2000 (AS-SPCC)*. Lake Louise, Alberta, Canada.

West, M. (1993). Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computer Science and Statistics* **24**, 325–333.

Wills, A., T. B. Schön and B. Ninness (2008). Parameter estimation for discrete - time nonlinear systems using EM. In: *Proceedings of 17th IFAC World Congress*. Seoul, South Korea.

Yeredor, A. (2000). The joint MAP-ML criterion and its relation to ML and to Extended Least-Squares. *IEEE Transaction on Signal Processing* **48**(12), 3484–3492.

Zhou, S. K., R. Chellappa and B. Moghaddam (2004). Visual Tracking and Recognition using appearance - adaptive Models in particle filters. *IEEE Transaction on Image Processing* **13**(11), 1491–1506.

Summary

Particle filtering/smoothing is a relatively new promising class of algorithms to deal with the estimation problems in nonlinear and/or non-Gaussian systems. Currently, this is a very active area of research and there are many issues that are not either properly addressed or are still open.

One of the key issues in particle filtering is a suitable choice of the importance function. The optimal importance function which includes the information from the most recent observation, is difficult to obtain in most practical situations. In this thesis, we present a new Gaussian approximation to this optimal importance function using the moment matching method and compare it with some other recently proposed importance functions.

In particle filtering/smoothing, the posterior is represented as a weighted particle cloud. We develop a new algorithm for extracting the smoothed marginal maximum *a posteriori* (MAP) estimate from the available particle cloud of the marginal smoother, generated using either the forward-backward smoother or the two filter smoother. The smoothed marginal MAP estimator is then applied to estimate the unknown initial state of a dynamic system.

There are many approaches to deal with the unknown static system parameters within particle filtering/smoothing set up. One common approach is to model the parameters as a part of the state vector. This is followed by adding artificial process noises to this model and then estimate the parameters along with the other state variables. Although this approach may work well in (certain) practical situations, the added process noises may result in a unnecessary loss of accuracy of the estimated parameters. Here we propose some new particle filtering/smoothing based algorithms, where we avoid any effect of the artificial dynamics on the estimate of the parameters.

Samenvatting

Particle filtering/smoothing is een relatief nieuw veelbelovende klasse van algoritmen om de schattingsproblemen bij niet -lineaire en/of niet-Gaussisch systemen aan te pakken. Op dit moment is dit een erg actief onderzoeksgebied en er zijn diverse aspecten hiervan die kregen geen voldoende aandacht of zijn ze nog niet opgelost.

Een centraal punt van aandacht bij de particle filtering is de juiste keuze van de importance functie. De optimale importance functie bevat informatie over de meest recente waarneming en is in het algemeen erg moeilijk te vinden. In dit proefschrift een nieuw Gaussisch benadering van de optimale importance functie is ontwikkeld en vergeleken met een paar andere recentelijk voorgestelde importance functies.

In particle filtering/smoothing, de posterior is weergegeven door een gewogen deeltjeswolk. Wij ontwikkelen nieuwe algoritme om de gesmoothed marginale maximale a posteriori (MAP) schatter van de beschikbare deeltjeswolk te extraheren. Het wordt voortgebracht door de forward-backward smoother of de twee filter smoother. Vervolgens is de gesmoothed marginale MAP schatter toegepast bij het schatten van de onbekende begintoestand van een dynamisch systeem.

Er zijn diverse manieren om de onbekende statisch systeem parameters binnen het kader van de particle filtering/smoothing te behandelen. Een standaard aanpak is het meenemen van de onbekende parameters binnen de toestandsvector als additionele componenten. Vervolgens wordt kunstmatige procesruis toegevoegd en de parameters worden mede geschat met de echte toestandsvectoren . Hoewel deze aanpak bevredigend werkt in (bepaalde) praktische situaties, de toegevoegde procesruis kan leiden tot onnodige vermindering van de nauwkeurigheid van de geschatte parameters. We formulieren hier een paar nieuwe algoritmen gebaseerd op particle filtering/smoothing waarbij mogelijke invloed van de kunstmatige dynamica op de schatting van de parameters wordt vermeden.

Acknowledgements

At the outset, I thank Prof. Arun Bagchi for accepting me as a PhD student in his group. I would like to express my sincere appreciation to my research supervisors, Prof. Arun Bagchi and Dr. Pranab K. Mandal for their active guidance, immense patience and encouragement throughout these past four years. I would also like to express my deepest sense of gratitude to Dr. Yvo Boers and Dr. J. N. Driessen for their enthusiastic involvement, kind cooperation and help extended to me throughout my research work of the thesis. I am also grateful to Prof. Shin Ichi Aihara for the fruitful collaboration and his critical support at the moment of writing this thesis.

I thank all the members of my promotion committee: Prof. Fredrik Gustafsson, Dr. J. N. Driessen, Prof. C. H. Slump, Prof. Anton A. Stoorvogel and Dr. Michel Vellekoop for their time and effort in reviewing a draft of this thesis and for their insightful comments and valuable suggestions.

I would like to cordially thank all the faculty members, the current, former and the visiting members of the Systems, Signals and Control group for their pleasant company. In particular, I would like to thank Marja Langkamp and Marjo Mulder for their ever willingness to help me in any administrative matter.

Also many thanks go to all my friends that I have met in Enschede and in Leuven. I owe a special note of gratitude to Arianto, Emad, Jaroslav, Ramakrishna, Hanumant, Pravin, Sachin and Ruta for their unhesitating support to me. In addition, I express my heartfelt thanks to Debasish, Sayan, Sunetra, Himadri, Arunasish, Mira and Nilanjan for their wonderful friendship and continuous inspiration from a distance.

I owe a debt of gratitude to Mrs. Pratima Mookerjee and Mrs. Arundhati Bagchi, whose support and encouragement made everything easier. Having their friendly association has been certainly a great privilege for us.

I would like to share this moment of happiness with my sister, sisters-in-law, brothers-in-law and their kids, Tunir, Tanuj, Aritra, Pallavi and Baisakhi, for their endless support and encouragement.

For me, the continuous support and encouragement of both my parents and parents-in-law have been indispensable. Words are not sufficient to express my gratitude for all the difficult decisions they had to make in their life on account of me and for being there always whenever I needed

them. Finally, Sangeeta has been an incredible source of support and inspiration. It would have been really impossible for me to accomplish this work without her love and encouragement. She is not just my wife, but my best friend and my strongest critic. I can not thank her enough for simply being the perfect companion one could ever wish for.

Saikat Saha

Enschede, August 2009