AN OVERVIEW OF EFFICIENT NONLINEAR FILTERING - FROM KALMAN FILTER TO PARTICLE FILTERS TO EIS

by

Weiwei Jiao

B.S. in Public Finance, Nanjing University of Finance & Economics, 2006

M.S. in Econometrics, Capital University of Economics & Business, 2011

M.S. in Mathematics, University of Arkansas at Little Rock, 2012

Submitted to the Graduate Faculty of the

Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This thesis was presented

by

Weiwei Jiao

It was defended on

December 2nd, 2015

and approved by

Kenneth D. Jordan, Richard King Mellon Professor and Distinguished Professor of Computational Chemistry, Department of Chemistry

Daniel S. Lambrecht, Assistant Professor, Department of Chemistry

Thesis Advisor: Jean-François Richard, Distinguished University Professor, Department of Economics

AN OVERVIEW OF EFFICIENT NONLINEAR FILTERING - FROM KALMAN FILTER TO PARTICLE FILTERS TO EIS

Weiwei Jiao, M.S.

University of Pittsburgh, 2015

Generally, in most applied fields, the dynamic state space models are of nonlinearity with non-Gaussian noise. However, as a famous and simple algorithmic filter, Kalman filter can only estimate linear system with Gaussian noise state space models. The Extend Kalman filter and the Unscented Kalman filter still have limitations and therefore are not accurate enough for nonlinear estimation. The Bayesian filtering approach which is based on sequential Monte Carlo sampling is called particle filters. Particle filters were developed and widely applied in various areas because of the ability to process observations represented by nonlinear state-space models where the noise of the models can be non-Gaussian. However, particle filters suffer from two long-standing problems that are referred as sample degeneracy and impoverishment. To fight these problems, resampling step is necessary. In this review work, a variety of resampling of particle filter methods as well as their characteristics and algorithms are introduced and discussed, such as Sampling-Importance resampling, Auxiliary particle filter, Optimal resampling and so on to combat against the sample degeneracy and impoverishment. Finally, efficient importance sampling, as a more accurate method, capable of estimating high-dimensional integration and carrying out global optimization, will be introduced and compared to particle filters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

PREFACE

I would like to first thank my advisor, Dr. Jean-François Richard, sincerely for his support and supervision. His profound knowledge, patient guidance, and deep tolerance were the biggest driving forces to make this thesis appear. I sincerely thank Dr. Kenneth Jordan and Dr. Daniel Lambrecht for serving as my committee members. I thank Dr. Kenneth Jordan and the program administrator Ms. Wendy Janocha for their continuous help in management during my study period in the computational modeling and simulation program. I would express my appreciation to Dr. Bing Liu, Dr. Zheng Han, Ms. Chengying Luo, Mr. Qingcheng Yang, Dr. Li Zhong, Ms. Jiayun Wang, Ms. Yingjun Su, and Mr. Hao Feng for their help, encouragement, and friendship during my study period at University of Pittsburgh. I also thank Dr. Tiancheng Li for spending time discussing particle filters with me and for providing me with his newly published papers.

My deepest thank to my husband Dr. Chenzhi Wang, my father Jie Jiao, my mother Chunling Wang, my father in law Shaoqing Wang, and my mother in law Shuling Qiu, for their love and support.

1.0 INTRODUCTION

It is well known that the Kalman filter (KF) is the best and simplest algorithmic filter for dynamic state estimation (Kalman). The KF is relatively easy to design and code. However, KF can only give the optimal estimation for systems with linear dynamics and additive Gaussian noise in the transition and the measurement functions. The dynamic models constituted by the transition functions and measurement functions are called the state space models (SSMs). SSM can estimate both process and observation errors though stochastic process. SSM provides a general framework for analyzing deterministic and stochastic dynamic systems. The SSM framework has been successfully applied in various areas to solve a broad range of problems in dynamic systems. In most cases, the SSMs are nonlinear models with non-Gaussian noise. Thus, it is impossible to get the accurate form of posterior probability density function (PDF). A number of nonlinear filtering methods have been developed to overcome the KF drawbacks.

Smith (1962) used Taylor series expansions to linearize a state space model at a working point with Gaussian noise, with the method called extended Kalman filter (EKF) [1]. Basically, the EKF simply perform linearization on all nonlinear transformations, and then replaces the nonlinear transformations by Jacobian matrices in the KF functions. However, EKF is hard to implement, hard to modify, and only dependable for systems that are mainly linear on the time scale of the modifications which come from the use of linearization [2]. The Unscented Kalman filter (UKF), proposed by Julier and Uhlman [3] [4], is a derivative-free alternative method to linearization. By using a statistical linearization method, UKF can easily approximate a probability distribution. The nonlinear functions are applied to each point, and then the

transformed points are computed in order to estimate the nonlinearly transformed mean and covariance [5]. Although this method does not require the dynamic system to be almost linear, it is still a linear approximation method with some weak points. First of all, the samples are not drawn at random. They are actually fixed with some specific properties such as having a given mean and covariance. Secondly, the samples are generated with weights. The summation of the weights is one, but it allows the weights to be positive or negative.

Sequential importance sampling (SIS) was first developed by Handschin and Dayne in the 1950s for the purpose of doing molecular simulations. Combining powerful sequential Monte Carlo sampling methods with Bayesian inference at an affordable computational cost, the concept called particle filter (PF) was introduced to work on a wide variety of nonlinear filtering problems for parameter estimation and state estimation [6]. It has been over two decades since the pioneering contribution of Gordon, Salmond and Smith (1993) was published [7], which is commonly regarded as the first instance of the modern Sequential Monte Carlo (SMC) algorithm. The name particle filter was first mentioned by Del Moral (1996) [8], also referred to sequential imputations by Liu and Chen (1996) [9], the survival of fittest and the likelihood weighting algorithm by Kitagawa (1996) [10], the Monte Carlo filter by Kitagawa[10], and the condensation filter by Isard and Blake (1998) [11]. To be consistent, in the following part of this thesis, the term particle filter will be used. PF is able to estimate the dynamic systems which have nonlinear state-space models and those which have non-Gaussian noise in the state space models. The advantage of the PF methods is that they can solve the difficult numerical integration problems. Therefore, PFs are extremely useful in diverse real-world applications, such as economics [12-14], finance [15], digital signal processing [16, 17], wireless

communications [18], automation and controls [19], navigation and tracking [20, 21], robotics [22], weather prediction, GPS [23], and geophysical information systems [24, 25].

Different from the UKF methods, PF explores the state space with randomly generated samples which are also referred as particles. Posterior probability is represented by samplers which are generated directly from the state space, and then the posterior is updated by including the new observations. By following the Bayesian principle, the particle system is appropriately placed, weighted, and propagated recursively [6]. When the number of the particles is very large, the distributions PDF of interest are approximated by the generated particles and the weights assigned to the particles. The PF approaches the optimal Bayesian estimation. There are many researchers who developed PF methods [26, 27].

However, PF suffers from two major problems: the first one is the sample degeneracy which was revealed by Doucet (1998) [28]; the second one is the sample impoverishment. The two problems have been a long-standing topic in the academic community. A variety of solutions have been proposed to combat these two main problems. In order to fight against sample degeneracy - the weakness of the particle filter, the regular PF is usually accompanied with the re-sampling approach. A lot of statisticians investigated and developed the so-called sampling-importance-resampling (SIR) for particle filter.

Nevertheless, the widely used re-sampling methods can lead to another issue - sample impoverishment, which was described in [29, 30]. Several researchers have developed methods to address the issue. Recently, a lot of work has been carried out to improve the performance of particle filters thanks to the fast development of high performance computers.

Different from the frame of the PF and SIR, a new method called efficient importance sampling (EIS) was developed by Zhang and Richard (2007) [31]. EIS is based on importance

sampling and minimizing the Monte Carlo variances. The advantages of this method include its ability to estimate high-dimensional interdependent integrals, a better accuracy on estimation, and the ability to perform global optimization. In each time, the EIS method is combined with likelihood evaluation therefore it can produce global approximations to targeted integrands. The EIS method has been applied for dynamic stochastic general equilibrium (DSGE) models [32, 33]. The EIS method was developed by a combination of the importance sampling and the auxiliary particle filter.

Table 1 compares the algorithms of the above mentioned methods [34].

In section 2, the SSM and all the associated notations will be introduced. The importance sampling, as the basic idea for estimating integrals which serve for PFs and EIS, will be described in this section.

In section 3, the basic algorithm of PF will be given firstly. To overcome the sample degeneracy drawback, several resampling methods for PF will be reviewed in detail. Moreover, several other resampling methods will also be presented to overcome another problem - the sample impoverishment, which is caused by the widely used resampling procedures.

In section 4, EIS filter will be presented with an example. And by using the same example, the effectiveness of the EIS filter and the PF will be compared.

Table 1. Comparison of algorithms for nonlinear filtering

Methods	EKF	UKF	PF &SIR	EIS
Statistics from one measurement period to the next	Use Taylor Series expansions to linear approximation the measurement functions	Use the "unscented transformation" to approximation the integrals	Use importance sampling & resampling to Monte Carlo sampling of the conditional density	Use importance sampling and regression to resampling to the Monte Carlo sampling of the conditional density
Accuracy of state vector estimate	Sometimes good but requires the models to be near linear	Provides a significant improvement to the EKF, but sometimes it does not	Optimal performance for low dimensional problems, suboptimal for high dimensions according to the computer speed	Optimal performance for low and high dimensional problems
Computational complexity	On the order of d^3 for estimating state vectors of dimension d	Roughly the same as the EKF	According to the algorithm of the PF	More complexity than PF

2.0 LIKELIHOOD EVALUATION IN STATE-SPACE AND IMPORTANCE SAMPLING

In order to describe efficient filters, the SSM will be defined at first. As described in the introduction chapter, the SSM, constituted by the transition functions and measurement functions, gives a useful general framework for analyzing dynamic models. The nonlinear filtering is a class of stochastic processes that widely exist in a variety of real-world fields. The solution of the continuous time filtering problem can be represented as a ratio of the two expectations of certain functions. The continuous-time SSM can be converted into a discrete-time state space model by sampling the outputs through discretization procedure. The SSM can be categorized to discrete or continuous time dynamic system. Basically, in the field of signal processing, most of the applications use discrete time dynamic system, while continuous time dynamic system has more applications in field of economics and finance.

2.1 THE STATE SPACE MODEL (SSM)

SSM consists of two equations:

One is the state transition equation:

$$s_t = g_t(s_{t-1}, u_t) \tag{1a}$$

Correspondingly, the transition probability density function is given by:

$$f(s_t \mid s_{t-1}, Y_{t-1}) \tag{1b}$$

Another one is the observation measurement function:

$$y_t = h_t(s_t, v_t) \tag{2a}$$

Correspondingly, the measurement probability density function is given by

$$f(y_t \mid s_t, Y_{t-1}) \tag{2b}$$

For easy notation, it is assumed that

$$f(s_t \mid s_{t-1}, Y_{t-1}) = f(s_t \mid s_{t-1}) \tag{1b}$$

$$f(y_t \mid s_t, Y_{t-1}) = f(y_t \mid s_t) \tag{2b}$$

Table 2. Notations for SSM

s_t	A $m \times 1$ vector of latent state variables at time t
S_t	$S_t \triangleq (s_0, s_1, ..., s_t)$, the history path of the state
y_t	A $n \times 1$ vector of observable variables at time t
Y_t	$Y_t \triangleq (y_1, ..., y_t)$, the history path of the observation
$g_t(\cdot)$	The state transition equation at time t
$h_t(\cdot)$	The observation equation at time t
u_t	Noise affecting the system dynamic equation $g_t(\cdot)$, at time t
v_t	Noise affecting the observation equation $h_t(\cdot)$, at time t
$s_t^{(i)}$	The state of particle i, at time t
$w_t^{(i)}$	The weight of particle i, at time t
N_t	The total number of particles at time t
$\delta_s(\cdot)$	The delta-Dirac measure located in s ($\delta(0) = 1; \delta(x) = 0$ for $x \neq 0$)
$N(.: a, b)$	Gaussian density with mean a and covariance b
$K_h(\cdot)$	A kernel function with bandwidth h

The assumption made here is that s_t is Markovian, i.e., its conditional probability density given the past states S_t depends only on s_t through the transition density $f(s_t|s_{t-1})$, and that the conditional probability density of y_t given the states S_t and the past observations Y_{t-1} depends only on s_t through the conditional likelihood $f(y_t|s_t)$. It is further assumed that the initial state s_0 is distributed according to a density function $f(s_0)$. Thus, the states and data may be sampled one by one by drawing random samples from Eq. (1b) and Eq. (2b), the algorithm is following:

Algorithm 1: Generate from a SSM

Initialization: Draw sample $\tilde{s}_0$ from $f(s_0)$

Draw sample $\tilde{y}_0$ from $f(y_0|\tilde{s}_0)$

For t=1 to T do

Draw sample $\tilde{s}_t$ from $f(s_t|\tilde{s}_{t-1})$

Draw sample $\tilde{y}_t$ from $f(y_t|\tilde{s}_t)$

End For

$\tilde{S}_T$ and $\tilde{Y}_T$ is a random draw from the transition and the observation densities functions.

Algorithm 1 simulates random data and it is a very important part of the particle filter algorithms which will be described later.

2.2 PREDICTION AND UPDATING

The filtering problem recursively solving the marginal posterior density $f(s_t | Y_t)$ which is called the period t filtering density can be determined by the recursive Bayesian estimation, which has two steps:

First step is to compute predictive density function in the period t, which is given by

$$f(s_t | Y_{t-1}) = \int f(s_t | s_{t-1}) f(s_{t-1} | Y_{t-1}) \, ds_{t-1} \tag{3}$$

Likelihood integral (just in the period t):

$$f(y_t | Y_{t-1}) = \int f(y_t | s_t) f(s_t | Y_{t-1}) \, ds_t \tag{4}$$

The likelihood function for the whole period:

$$f(Y_T) = \prod_{t=1}^{T} f(y_t | Y_{t-1}) \tag{5}$$

Second step is to compute filtering density in the period t ,

$$f(s_t | Y_t) = \frac{f(y_t | s_t) f(s_t | Y_{t-1})}{f(y_t | Y_{t-1})} \tag{6}$$

Here, we initialize $f(s_0 | Y_0) = f(s_0)$. In Eq. (3) and Eq. (4), the integration of unknown functions and probably high-dimensional functions is required. However, this integration is very difficult to compute. Therefore, this difficulty makes analytic optimal solutions such as the Kalman filter intractable. A convenient solution for this problem is the importance sampling (IS). In the next section, the IS method will be briefly reviewed and discussed.

2.3 IMPORTANCE SAMPLING

The PF, SIR, and EIS are all based on the importance sampling (IS), which is a general technique for estimating the properties of particular distributions. IS only needs the samples generated from a different distribution of interest. For example, we want to get the expectation of $p(s)$, which is the density of interest. But it is impossible to get the expectation from the integral $\int f(s)p(s)ds$. And we are unable to sample from a required distribution $p(s)$. Assuming that the density $q(s)$ roughly approximates the $p(s)$, then we can use a trick that will allow us to sample from a known distribution, $q(s)$. The trick is that we can generate the samples from the available distribution, e.g., a Gaussian distribution, and then re-weight the samples to approximate it. Then the expectation over $p(s)$ can be computed to yield the following approximation:

$$\begin{aligned} \hat{f} &= \underset{p(s)}{E}(f(s)) = \int f(s)p(s)ds = \int f(s)(\frac{p(s)}{q(s)})q(s)ds \\ &= \underset{q(s)}{E}\{f(s)(\frac{p(s)}{q(s)})\} \end{aligned} \tag{7}$$

$\frac{p(s)}{q(s)}$ is called the importance weight. This forms the basis of Monte Carlo importance sampling which uses the weighted sum of a set of samples from $q(s)$ to approximate Eq. (7):

$$\hat{f} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p(s^i)}{q(s^i)}f(s^i) \tag{8}$$

We draw N random samples from $\frac{p(s)}{q(s)}$, and assign a positive weight to each of the random points. Let $w^i = \frac{p(s^i)}{q(s^i)}$. The expectation can thus be estimated using a weighted function:

$$\hat{f} \approx \frac{\sum_{i=1}^{N} w^i f(s^i)}{\sum_{j=1}^{N} w^j} \tag{9}$$

and its variance is given by:

$$\mathrm{var}(\hat{f}) = (w^j)^2 \left[\int \frac{(f(s)p(s))^2}{q(s)} dx - E_p\{f(s)\} \right] \tag{10}$$

where s^i is drawn from the known density $q(s)$. If $p(s) = q(s)$, the variance is minimized to zero [35]. An easy choice for $q(s)$ is linear piecewise functions which can simulate $p(s)$ with small variance. However it's not easy to code and its computational complexity is high. There are many potential choices for $q(s)$ leading to various integration and optimization algorithms, as shown in the summary provided in [36] by Del Moral, Doucet et al. (2006). In general, the density function of $q(s)$ should have relatively heavy tails so that it is insensitive to the outliers. In statistics, the heavy-tailed distributions are the probability distributions whose tails are not exponentially bounded. In other words, it's better to draw samples from a known heavy-tailed distribution. It is impossible to generate the sampling by using $p(s)$ because the density $p(s)$ of interest is generally unknown. To overcome this drawback, the IS heavily depends on the information of $q(s)$, which is irrelevant with $p(s)$. Several proposed IS methods are summarized in Table 3 [30].

The particle filter approach is based on sequential importance sampling (SIS), which is an IS method implemented by using the recursive Bayesian interference. A weight update equation for different proposal kernels can be computed by specifically generated SMC samplers from a sequence of probability distributions [36]. In EIS, the exponential family of distributions will be adopted for the kernel function.

Table 3. Advanced important sampling methods

Annealed importance sampling	Radford, 2001
Bayesian importance sampling	Rasmussen & Ghahramani, 2003
Adaptive importance sampling	Liu & West, 2001
Numerically accelerated importance sampling	Koopman, Lucas, & Scharth, 2011
Nonparametric importance sampling	Neddermeyer, 2011
Block sampling	Doucet, Briers, & Senecal, 2006
Markov Chain Monte Carlo (MCMC) sampling	Gilks & Berzuini, 2001; Del Moral et al., 2006
Factored sampling	Banerjee & Burlina, 2010; Isard & Blake, 1998
Multiple stages of important sampling	Li, Ai, Yamashita, Lao, & Kawade, 2008

3.0 PF AND SIR

3.1 PF

The particle in PF does not mean a minute portion, a piece, a fragment, or an amount. It means a sample or an individual which is drawn from a known density function. Basically, the particle filters are used to capture the distribution of the state probability by using a group of random particles with corresponding weights, then the particle filter in Eq. (6) becomes

$$\hat{f}(s_t \mid Y_t) \approx \sum_{i=1}^{N_t} w_t^i \cdot \delta(s_t - s_t^i) \tag{11}$$

Where $\left\{s_t^i\right\}_{i=1}^N$ denotes a vast body of particles drawn from a period t IS density, and $\left\{w_t^i\right\}_{i=1}^N$ denotes the respective IS weights. The weights are normalized to be sum to one ($\sum_{i=1}^N w_t^i = 1$) [37]. Then

$$w_t^i \propto w_{t-1}^i \frac{f(y_t \mid s_t^i) f(s_t^i \mid s_{t-1}^i)}{q(s_t^i \mid s_{t-1}^i, y_t)} \tag{12}$$

Here we assume that $q(s_t^i \mid s_{t-1}^i) = q(s_t^i \mid s_{t-1}^i, y_t)$, then Eq. (12) becomes

$$w_t^i \propto w_{t-1}^i \frac{f(y_t \mid s_t^i) f(s_t^i \mid s_{t-1}^i)}{q(s_t^i \mid s_{t-1}^i)} \tag{12}$$

It can be shown that when $N \to \infty$, the approximation of particle filtering in Eq. (11) approaches the true filtering density $f(s_t|Y_t)$, defined above in Eq. (6). The algorithm of SIS is generating particles according to a known density function $q(s_t^i|s_{t-1}^i, y_t)$, then assigning weights according to Eq. (12) to the corresponding particles. A pseudo-code description of this algorithm is given by Algorithm 2 as following:

Algorithm 2 : Basic PF(SIS)

$$\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N\right] = basicPF\left[\left\{s_{t-1}^i, w_{t-1}^i\right\}_{i=1}^N, y_t\right]$$

For i=1 to N do

Draw $\tilde{s}_0^i$ from $q(s_0|y_0)$

Assign initial weights

$$\tilde{w}_0^i = \frac{f(y_0|\tilde{s}_0^i) f(|\tilde{s}_0^i)}{q(\tilde{s}_0^i|y_0)}$$

End for

For t=1 to T do

For i=1 to N do

Draw sample $\tilde{s}_t$ from $q(\tilde{s}_t^i|\tilde{s}_{t-1}^i, y_t)$

Compute weights according to Eq. (12)

End For

For i=1 to N

Normalize weights

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^{N} \tilde{w}_t^j}$$

End for

Compute filtering estimate according to Eq. (11)

End for

The basic PF is also called SIS, which does not depend on the underlying Markov chain. Instead, the SIS creates an importance sample, which consequently improves the efficiency. The SIS can be also used in a non-Bayesian computation, for example, to estimate the likelihood value. However, the importance weights may have large variances, resulting in an inaccurate estimate. A common problem with the SIS particle filter is the sample degeneracy phenomenon which has been defined and demonstrated by Doucet, Godsill et al. (2000) [27].

In practice, after a few iterations of the algorithm, only one particle's weight is almost close to one, and all other particles' weights are very close to zero. This is namely sample degeneracy. Doucet proofed that the degeneracy phenomenon is not possible to avoid because the variance of the importance weights can increase over time. This degeneracy implies that a considerable computational cost is used for updating particles whose contribution to the approximation is nearly zero. This is an inherent feature of the SIS. To overcome sample degeneracy, the standard PF is usually used with the resampling procedure. This procedure is seen to be referred to the so-called Sampling-Importance Resampling (SIR) or Sequential Importance Sampling and Resampling (SISR), through which the particles are forced to the areas

of high likelihood by multiplying the high-weighted particles with their respective weights, while low-weighted particles are discarded in this procedure.

3.2 SIR

The sample degeneracy causes the obtained estimates not to be accurate and to have unacceptably large variances. With resampling, such shortcomings can be prevented. Consequently, resampling methods have been extensively researched. A variety of resampling schemes have been proposed by Kitagawa (1996) [10], Beadle and Djuric (1997) [38], Liu and Chen (1998) [39], Carpenter, Clifford et al. (1999) [40], and Liu, Chen et al. (2001) [41]. The surveys of a number of resampling methods can also be found in many papers: Bolic, Djuric et al. (2004) [42], Douc and Cappe (2005) [43], Hol, schon et al. (2006) [44], and Doucet and Johansen (2009) [45].

Next, these resampling methods will be reviewed and discussed. With the development of PFs, in this thesis, some of the new methods will be overviewed. These new methods were developed by Li, Sattat et al. (2012) [29], Li, Sun et al. (2014) [30], Li, Bolic et al. (2015) [46], and Li, Villarrubia et al. (2015) [47]. In those papers, another problem called sample impoverishment was introduced.

The resampling is used to eliminate particles with small importance weights and to duplicate the particles of large weights. Almost all of these resampling methods of PF are based on three steps: particle generation, weight computation, and resampling. Instead of generating particles and assigning with corresponding computed weights, the three steps of resampling

procedure will generate another set of particles with weights, that can overcome the sample degeneracy problem.

3.2.1 Multinomial Resampling

The Multinomial resampling method was developed by Gordon, Salmond et al. (1993) [7] based on an idea at the heart of the bootstrap method. The main idea of multinomial resampling is to generate N random numbers $\left\{u_t^i\right\}_{i=1}^N$ from the uniform distribution on $(0,1]$, which are independent, then to use the random numbers to select samples. In the nth selection, the sample s_t^i is chosen when the following condition is meet:

$$\Omega_t^{i-1} < u_t^n \le \Omega_t^i \tag{13}$$

where

$$\Omega_t^i = \sum_{j=1}^{i} w_t^j \tag{14}$$

Then the probability of choosing particle s_t^i is the same as that of u_t^n being in the interval bounded by the cumulative sum of the normalized weights as shown in Eq. (13). The estimates from this resampling method meet the unbiasedness condition.

Here, only part of the resampling algorithm will be described:

Algorithm 3 : Multinomial resampling

$$\left[\left\{s_t^i\right\}_{i=1}^N\right] = Multinomial\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N, N\right]$$

$$\left[\left\{\Omega_t^i\right\}_{i=1}^N\right] = CumulativeSum\left[\left\{w_t^i\right\}_{i=1}^N\right]$$

At time t

$w_t^i = \frac{w_t^i}{\sum_{j=1}^{N} w_t^i}$ % Normalize weights

$\Omega_t^i = \sum_{j=1}^{i} w_t^j$;

Index=zeros(1,N);

$i = 0$;

While $i < N$

$i = i + 1$;

Draw u_t^i from uniform distribution on $(0,1]$;

$j = 1$;

While $\Omega_t^j < u_t^i$

$j = j + 1$;

End

Index(i)=j;

end

Since the sampling of each particle is random, the upper and lower limits of the number of the times that a given particle is resampled are zero and N_t, respectively. This yields the maximum variance of the resampled particles. The computational complexity of the multinomial resampling is of order $O(NM)$. It is known that multinomial resampling is not efficient in

finding Eq. (13). The computational complexity of finding Eq. (13) is of order $O(N)$. A faster search method called binary search has the computational complexity of order $O(\log N)$. The binary search is well known in computer science and here is used to execute the search of n in Eq. (13), therefore reducing the computational complexity from N to $\log(N)$. In the next step, the systematic resampling and the stratified resampling which reduce the variance of the resampled particles will be introduced. These two methods introduce the idea of strata into the multinomial resampling method.

3.2.2 Systematic Resampling

The Systematic resampling method was first proposed by Kitagawa (1996) and further discussed by Carpenter, Clifford et al. (1999) [10, 40]. It is the basic and simplest resampling method. The core idea is to reset each sample with an equally fixed weight $w_t^{*i} = \frac{1}{N}$ in each time. The algorithm is given as following:

Algorithm 4: Systematic Resampling(SIR)

$$\left[\left\{s_t^{*i}\right\}_{i=1}^{N}\right] = Systematic\left[\left\{s_t^{i}, w_t^{i}\right\}_{i=1}^{N}\right]$$

Initialization

For i=1 to N do

Draw $\tilde{s}_0^i$ from $q(s_0 | y_0)$

Assign initial weights

$$\tilde{w}_0^i = \frac{f(y_0 \left| \tilde{s}_0^i \right.) f(\left| \tilde{s}_0^i \right.)}{q(\tilde{s}_0^i \left| y_0 \right.)}$$

End for

Resample $\left\{ s_0^i, w_0^i \right\}_{i=1}^N$ to get the new samples $\left\{ s_0^{*i}, w_0^{*i} = \frac{1}{N} \right\}_{i=1}^N$

For t=1 to T do

For i=1 to N do

Set $\left\{ s_t^i \right\}_{i=1}^N = \left\{ s_t^{*i} \right\}_{i=1}^N$

Draw sample s_t^i from $q(s_t^i \left| s_{t-1}^i, y_t \right.)$

Compute weights according to Eq. (12)

End For

Resample $\left\{ s_t^i, w_t^i \right\}_{i=1}^N$ to get the new samples $\left\{ s_t^{*i}, w_t^{*i} = \frac{1}{N} \right\}_{i=1}^N$

End for

Compute filtering estimate according to Eq. (11)

End for

The Stratified resampling developed by Kitagawa (1996) [10] also divides the whole population of particles in to N subpopulations. The random numbers $\left\{ u_t^i \right\}_{i=1}^N$ are drawn independently from uniform distribution on each of N disjoint subintervals $(0, \frac{1}{N}] \cup ... \cup (1 - \frac{1}{N}, 1]$ of $(0,1]$. And the bounding method on the cumulative sum of normalized

weights as shown in Eq. (13) is used. This procedure is same as that of the multinomial resampling method.

The systematic resampling and stratified resampling methods can be implemented simply in $O(N)$ time and minimizing the Monte Carlo variation. Thus, it is straightforward to conclude that this approach is unbiased. Due to the smaller number of random numbers generated, the systematic method is computationally more efficient than the stratified method.

3.2.3 Residual Resampling

The Residual resampling developed by Beadle and Djuric (1997) [38] is an alternative method to algorithmic systematic resampling. It contains two steps of resampling. The first step is to find which particle's weight is bigger than $\frac{1}{N}$, and the second step is to sample randomly using the remaining particles whose weights are smaller than $\frac{1}{N}$ (referred to as *residuals*). For the first step, set $\tilde{N}_t^i = \lfloor N w_t^i \rfloor$, and this step is called the deterministic replication part. The second step is residual resampling, setting the residual of the weight as:

$$\overline{N}_t^i = w_t^i - \frac{\tilde{N}_t^i}{N} \tag{15}$$

Then, the new particles are drawn by calling the multinomial resampling program with the parameters $\left(\overline{N}_t^i, R_t\right)$. That is $\left[\left\{s_t^i\right\}_{i=\tilde{N}_t+1}^{N}\right] = Multinomial\left[\left\{s_t^i, \overline{N}_t^i\right\}_{i=1}^{N}, R_t\right]$. The total number of replicated particles in the first stage is $N_t = \sum(\tilde{N}_t^i)$, and in the second step it is $R_t = N - \tilde{N}_t$. In

this sampling method, the resampling step will be introduced in algorithm 5, while the other part is the same as systematic resampling.

Algorithm 5 : Residual resampling

$$\left[\left\{s_t^i\right\}_{i=1}^N\right] = Residual\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N, N\right]$$

At time t

$$w_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$$ % Normalize weights

Index=zeros(1,N);

% step 1 Deterministic replication of particles

$$\left[\left\{\tilde{s}_t^i\right\}_{i=1}^N, N\right] = Replication\left[\left\{s_t^i, \tilde{N}_t^i\right\}_{i=1}^N\right]$$

$$\tilde{N}_t^i = \lfloor N w_t^i \rfloor$$

$$N_t = \sum(\tilde{N}_t^i)$$

$R_t = N - N_t$

$i = 1;\ j = 0$

while $j < N$

$j = j + 1$;

$cnt = 1$;

while $cnt \le \tilde{N}_t^j$

Index(i)=j;

$i = i + 1; cnt = cnt + 1;$

End

End

% step 2 resampling from Multinomial resampling

$$\overline{N}_t^i = w_t^i - \frac{\tilde{N}_t^i}{N}$$

For i=1 to M do

$$\overline{N}_t^i = \overline{N}_t^i \times \frac{N}{N - \tilde{N}_t}$$

End

$$\left[\left\{ s_t^i \right\}_{i=N_t+1}^{N} \right] = Multinomial \left[\left\{ s_t^i, \overline{N}_t^i \right\}_{i=1}^{N}, R_t \right]$$

$$\left[\left\{ \Omega_t^i \right\}_{i=1}^{N-N_t} \right] = CumulativeSum \left[\left\{ \overline{N}_t^i \right\}_{i=1}^{N-N_t} \right]$$

The algorithm $\left[\left\{ \tilde{s}_t^i \right\}_{i=1}^{N}, N \right] = Replication \left[\left\{ s_t^i, \tilde{N}_t^i \right\}_{i=1}^{N} \right]$ will be used in other resampling methods.

From the above algorithm, it is seen that for the residual resampling, the ith particle is resampled $\tilde{N}_t^i + R_t^i$ times, where $\tilde{N}_t^i$ is the number of replications from the first step, and R_t^i is the number of replications from the second step. Since the residual resampling has two steps, the computational complexity is of order $O(N_t) + O(R_t)$ time. The aim of this program is to break

the empirical cumulative distribution function up into N components, each of which is then sampled once.

The first step describes a deterministic replication, so the variation of the number of times a particle is resampled is only attributed to the second step. Thus, if the multinomial resampling method was used in the second step, the upper limit of the number of times that the ith particle is resampled is $\tilde{N}_t^i + R_t^i$, and the lower limit is $\tilde{N}_t^i$.

3.2.4 Residual systematic resampling

These resampling methods for PF are probably the best known and mostly used. They have been changed and developed in many ways. For example, within the second step of the residual resampling grogram, we can also use systematic resampling or stratified resampling programs to resample. If in the second step, the systematic resampling method is used to resample, the new method is called residual systematic resampling (RSR) which was described by Bolic, Djuric et al. [42, 48]. As described in multinomial resampling method whose sample is from the normalized fractions, the computational time is more complex than other resampling methods. Different from residual resampling which proceeds in two separate loops, RSR proceeds in only one loop with the integer replication. And there is no additional procedure required for the residuals. Thus, the computational complexity of RSR is of order $O(N)$. The idea of RSR is identical with the idea which is used in systematic resampling method. That is to accumulate the fractional donations of each particle in the searching list until it is large enough to generate a sample. The algorithm of the resampling part of RSR is as following:

Algorithm 6 : RSR

$$\left[\left\{\tilde{s}_t^i\right\}_{i=1}^N\right] = RSR\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N, N\right]$$

At time t

Draw a random number Δu form uniform distribution $U(0, \frac{1}{N}]$

$$w_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^i} \quad \% \text{ Normalize weights}$$

For i= 1 to N do

$$\tilde{N}_t^i = \lfloor N \times (w_t^i - \Delta u) \rfloor + 1$$

$$\Delta u = \Delta u + \frac{\tilde{N}_t^i}{N} - w_t^i$$

End

$$\left[\left\{\tilde{s}_t^i\right\}_{i=1}^N, N_t\right] = Replication\left[\left\{s_t^i, \tilde{N}_t^i\right\}_{i=1}^N\right]$$

3.2.5 Branch-kill resampling

All of the resampling methods introduced above generate particles with the fixed-size N on every time step by forcing the particle size to be a constant N . If the size is allowed to change, there are simple ways to generate particles in parallel and just in one loop. Parallel algorithm dramatically improves the speed of computation, thanks to the development of the high

performance computer, which allows parallel algorithms be adopted in PF. There is a method called the branch-kill procedure (Budhiraja, Chen et al. 2007) [49] or branching (Crisan and Lyons 1999) [50]. The algorithm is given by:

Algorithm 7: Branch-kill resampling

$$\left[\left\{\tilde{s}_t^i\right\}_{i=1}^{N}\right] = Branch-kill\left[\left\{s_t^i, w_t^i\right\}_{i=1}^{N}, N_{re}\right]$$

At time t

For i=1 to N do

Draw a random number Δu form uniform distribution $U(0, \frac{1}{N_{re}}]$

$$\tilde{N}_t^i = \lfloor N_{re} \times w_t^i \rfloor$$

If $\left(N_{re} \times w_t^i - \tilde{N}_t^i\right) \geq \Delta u$

$$\tilde{N}_t^i = \tilde{N}_t^i + 1$$

End

End

$$\left[\left\{\tilde{s}_t^i\right\}_{i=1}^{N}, N_t\right] = Replication\left[\left\{s_t^i, \tilde{N}_t^i\right\}_{i=1}^{N}\right]$$

In this method, the number of repeat generated particles of s_t^i is equal to $\tilde{N}_t^i = \lfloor N w_t^i \rfloor + 1$ with probability p or equal to $\tilde{N}_t^i = \lfloor N w_t^i \rfloor$ with the probability $1-p$. Where, $p = N w_t^i - \lfloor N w_t^i \rfloor$,

and N_{re} is the real number of generated particles in the last step. There is another approach is called rounding-copy resampling which was described by Li, Sattar et al. (2013) [51]. Just as its name implies, in this method, $\tilde{N}_t^i$ is the rounding result of Nw_t^i. These two parallel resampling methods do not need any additional operation and their samples are unbiased with a varying particle size. The main ideas of the RSR method, branch-kill resampling method, and rounding-copy resampling method come from the residual resampling method. Thus, their lower and upper limits of the repeat time of the i th particle are same: $\lfloor Nw_t^i \rfloor$ and $\lfloor Nw_t^i \rfloor + 1$, respectively.

The following Table 4 summarizes the characters of these resampling methods:

Table 4. A comparison of traditional resampling methods

Resampling method	Computational time	Number of random numbers	Fixed sample size	Lower limit	Upper limit
Multinomial resampling	$O(MN)$	N	YES	0	N
Systematic resampling	$O(N)$	1	YES	$\lfloor Nw_t^i \rfloor$	$\lfloor Nw_t^i \rfloor + 1$
Stratified resampling	$O(N)$	N	YES	0	$\lfloor Nw_t^i \rfloor + 2$
Residual resampling	$O(N)$	R_t	YES	$\lfloor Nw_t^i \rfloor$	$\lfloor Nw_t^i \rfloor + R_t$
Residual systematic resampling	$O(M) + O(R_t)$	1	YES	$\lfloor Nw_t^i \rfloor$	$\lfloor Nw_t^i \rfloor + 1$
Branch-kill resampling	$O(N)$	N	NO	$\lfloor Nw_t^i \rfloor$	$\lfloor Nw_t^i \rfloor + 1$
Rounding-copy resampling	$O(N)$	0	NO	$\lfloor Nw_t^i \rfloor$	$\lfloor Nw_t^i \rfloor + 1$

The resampling methods addressed so far are based on an approach where all the particles are sampled in the same way. This entails yielding relatively similar resampling results. For all of the methods, the condition of unbiasedness is satisfied, and the resampled particles are equally weighted. In the following content, the methods whose resampling is realized without attempting to satisfy the conditions of unbiasedness and equal-weighting are discussed. This may entail risks of which practitioners must be aware. After that, several new methods will be introduced, as well as a new problem.

3.2.6 Auxiliary Particle Filter

The auxiliary resampling particle filter (APF) method was introduced by Pitt and Shephard (1999) in [52] as a variant of the standard PF resampling filter. This filter introduced an importance density $q(s_t^i, i \mid s_{t-1}^i, y_t)$, which samples the particle with another index j_i according to the empirical distribution on the stochastic processes. Then the samples at time $t-1$ should be $s_{t-1}^i = \tilde{s}_{t-1}^{j_i}$, and the weights are assigned. The algorithm becomes:

Algorithm 8: Auxiliary Particle Filter

$$\left[\left\{\tilde{s}_t^i, \tilde{w}_t^i\right\}_{i=1}^N\right] = APF\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N, j\right]$$

Initialization

For i=1 to N do

Draw $\tilde{s}_0^i$ from $q(s_0 \mid y_0)$

Assign initial importance weights

$$\tilde{w}_0^i = \frac{f(y_0 \left| \tilde{s}_0^i \right.) f(\left. \tilde{s}_0^i \right|)}{q(\tilde{s}_0^i | y_0)}$$

End for

For t=1 to T do

Select N particle indices $j_i \in \{1,...,N\}$ according to weights

$$\left\{ v_{t-1}^i \right\}_{i=1}^N$$

For i=1 to N do

Set $s_{t-1}^i = \tilde{s}_{t-1}^{j_i}$

And set first stage weights

$$u_{t-1}^i = \frac{w_{t-1}^{j_i}}{v_{t-1}^{j_i}}$$

End for

For i=1 to N do

Draw sample $\tilde{s}_t^i$ from $q(\tilde{s}_t^i | s_{t-1}^i, y_t)$

Compute weights according to

$$w_t^i \propto u_{t-1}^i \frac{f(y_t \left| \tilde{s}_t^i \right.) f(\tilde{s}_t^i | s_{t-1}^i)}{q(s_t^i | s_{t-1}^i, y_t)}$$

End For

Normalize weights

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^{N} \tilde{w}_t^j}, \quad i = 1, ..., N \ .$$

End for

Compute filtering estimate according to Eq. (11)

End for

The previous methods were aimed primarily at improving the proposal distribution for the new state at the time instant t. Compared with these resampling methods above, the APF filter generates points form the sample at the time $t-1$, which are more likely to be close to the true state by being conditioned on the current measurement. In other words, APF can be viewed as resampling based on some point estimates u_{t-1}^i at the previous time step, represented as $f(s_t^i \left| s_{t-1}^i \right.)$. Thus, the APF filter is very useful when the noise is small because $f(s_t^i \left| s_{t-1}^i \right.)$ can be well described by u_{t-1}^i. However, when the process noise is very large, the APF resamples will be based on a poor approximation of $f(s_t^i \left| s_{t-1}^i \right.)$ because a single point can not describe the $f(s_t^i \left| s_{t-1}^i \right.)$ well. Thus, the APF filter is not fit for estimating a large noise state space model.

Several improvements were proposed to reduce its variance by Pitt and Shephard (2001) and Whiteley and Johansen (2010) [53, 54] in their summary of the APF method. Next, two of these methods will be introduced. One of the methods is called auxiliary marginal particle filter which was described by Klaas, De Freitas, et al. (2012) [55]. As we know that most particle filtering methods rely on a numerical approximation to evaluate the integrals Eq. (3) and Eq. (4), it differs in some details with the original formulation. Specially, the predictive density function Eq. (3) and the likelihood integral Eq. (4) are not obtained by approximating the predictive

distribution by drawing particles from their density functions. However, if Eq. (3) and Eq. (4) are obtained by doing so, this approach will have some difficulties. In APF method, an importance correction is adopted when this filter is used to approximate the Eq. (6). Thus the algorithm of the APF resampling method with this approach has a computational complexity $O(N^2)$, while most other particle filters lead to algorithms of complexity $O(N)$.

Another efficient APF method is the stratified auxiliary particle filter which reduces the variance. The method was first proposed by Karlsson and Bergman (2000) [56]. This method first draws each particle s_t^i by conditional density function $q(s_t^i \left| s_{t-1}^i, y_t \right.)$ instead of by randomly sampling a value to s_t^i. It evaluates one importance weight for every possible value of s_t^i, yielding a set of $N \times M$ weighted sample points. The resultant distribution on $\{1,2,...,N\} \times \{1,2,...,M\}$ was drawn N times in the next step of SIR algorithm - resampling step. This assignment can be performed with a low variance resampling mechanism.

3.3 SEVERAL NEW RESAMPLING METHODS TO FIGHT SAMPLING IMPOVERISHMENT

As described above, the goal of the resampling step is to reduce the effects of the degeneracy problem. However, the resampling step brings in other practical problems. First, because all the samples must be combined, it is hard to execute parallel computing. Second the resampling step has to choose the particles with high weights many times. Therefore, a high likelihood will be obtained by multiplying high weighted particles and their respective weights. This resampling procedure results in a loss of diversity among the particles. That means new samples will contain

many repeated points. In other words, a few particles with significant weights are repeated many times while most other particles with small weights are abandoned during the resampling process. This problem is known as “sample impoverishment”. Specially, in the case of very small noise in process, all particles will collapse to a single point after just a few iterations. In addition, any smoothed estimates, which are based on the paths of particles, will degenerate because the diversity of the particle paths is reduced.

Figure 1 shows the relationship between degeneracy and impoverishment. The weights of the particles are represented by circle size; after generic resampling, jointed circles share the same state as shown in the lower row of Figure 1 In the resampling procedure, only the large weighted particles (the red particles shown in Figure 1) are resampled. While the other particles with small weights (the blue particles shown in Figure 1) are discarded.

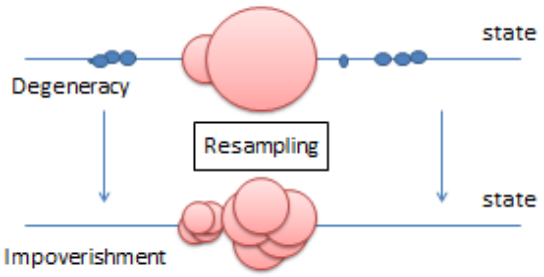

Figure 1. Trade-off between sample degeneracy and impoverishment

From Figure 1, sample degeneracy is obviously the result of particles distributed in various places (some with high weights, and some with low weights), while after resampling sample impoverishment can be viewed as particles just distributed in narrow region with similar weights. The red part of the particles with large weights which will be generated repeatedly more

times while the blue part of the particles will be ignored after resampling. The resampling step directly leads to the sample degeneracy problem, which will transfer to the sample impoverishment problem. Moreover, if the resampling is unbiased, a more severe degeneracy problem will lead to a more severe impoverishment problem.

To deal with the sample impoverishment problem, the trade-off between degeneracy and impoverishment should be balanced. That means when and how to apply resampling should be considered. The only method is to set up a pre-specified threshold at deterministic steps, i.e. to perform resampling only when the variance of the non-normalized weights is under the threshold. If particles have un-normalized weights with a small variance, the resampling step might be unnecessary. This is often evaluated by investigating the variability of the weights using the Effective Sample Size (ESS) criteria which was introduced by Kong, Liu et al. (1994) [57]. In some papers, the notation $\widehat{N_{eff}}$ is used as ESS instead.

$$\widehat{N_{eff}} = ESS = \left(\sum_{i=1}^{N} \left(w_t^i \right)^2 \right)^{-1} \tag{16}$$

Several resampling schemes will be introduced to alleviate impoverishment.

3.3.1 Optimal resampling

We consider the compound sampling methods which are based on grouping the particles by using pre-specified criterion before performing resampling. The groups are not overlapped, representing a part of the whole particle set. Particles with similar weights are organized to the same group by using weight-based thresholds as the criterion for grouping. Resampling is then

executed for each group in different ways. The application of compound resampling aims to reduce the resampling time and to preserve particle diversity.

In group based resampling, particles are put into different groups using the thresholds based on weight. Different sampling strategies can be adopted for each group to have more flexibility. The threshold can be deterministic or dynamic, and one can have one or a number of thresholds.

The optimal resampling was developed by Fearnhead and Clifford (2003) [58] who automatically sets a threshold value c_t, which has a unique solution of

$$N = \sum_{i=1}^{I} \min(\frac{w_t^i}{c_t}, 1) \tag{17}$$

where $I < N$. All the particles whose weights are higher than this threshold are completely preserved instead of being replicated. Therefore, multiple copies of these particles do not exist in the final set of N particles. Resampling are performed on the other particles using a probability according to their weights and assigned them with weight c_t. It is seen that the resampled particles do not have equal weights. The merit of the method among the unbiased resampling methods is its optimization in terms of minimizing the squared error-loss function

$$E\left(\sum_{i=1}^{I} \left(\tilde{w}_t^i - w_t^i \right)^2 \right) \tag{18}$$

Where $\tilde{w}_t^i$ is the new weight of s_t^i when it is resampled; otherwise, $\tilde{w}_t^i$ is equal to zero. Optimal resampling is suitable for PF that uses increased number of propagated particles. And this method reduces the number to $I < N$. A drawback is that it needs to calculate c_t in each iteration. In addition, the resampled particles may still have degeneracy issue because of the high variance of the weights. The algorithm is given:

Algorithm 9: Optimal resampling

$$\left[\left\{\tilde{s}_t^i, \tilde{w}_t^i\right\}_{i=1}^{N}\right] = Optimal\left[\left\{s_t^i, w_t^i\right\}_{i=1}^{I}, N\right]$$

Calculate c_t according to Eq. (17)

$i = 0; h = 0$

For i=1 to I do

If $w_t^i \geq c_t$

$i = i + 1$

$\tilde{s}_t^i = s_t^i$ and $\tilde{w}_t^i = w_t^i$

Else

$A^h = s_t^i$ and $B^h = w_t^i$

End

End

$N_1 = i$

$$\left[\left\{\tilde{s}_t^i\right\}_{i=N_1+1}^{N}\right] = Stratifiedresample\left[\left\{A^r, B^r\right\}_{r=1}^{h}, N - N_1\right]$$

For i = $N_1 + 1$ to N do

$\tilde{w}_t^i = c_t$

End

There are similarities between optimal resampling, rejection control resampling which was introduced by Liu, Chen et al. (1998) [59], and partial rejection control resampling which was described by Liu, Chen et al. (2001) [60]. Rejection control resampling method computes a control threshold c_t, which can be assigned in advance by the median or a quantile of the weights, and the ith particle is accepted with a probability given by

$$p = \min(\frac{w_t^i}{c_t}, 1) \tag{19}$$

In the partial rejection control method, the particles whose weights are larger than or equal to c_t are automatically accepted, while the other particles are accepted with probability p. So this method combines the rejection method and importance sampling. The rejected particles are replaced by the ones regenerated from previous time instances. An accepted particle s_t^i is reweighted with a new weight $\max\left(c_t, w_t^i\right)$. The difference of the two forms of rejection control resampling are basically about how far one goes back to regenerate particles. The rejection control resampling goes back to the earliest time, i.e. $t = 0$. While the partial rejection control resampling regenerates particles from $t-1$ to save computational time. Because of the undetermined execution time and high memory demand, these methods cannot be considered for real-time implementation.

3.3.2 Reallocation resampling

This part will introduce a fixed threshold resampling method. Reallocation resampling was proposed in Liu, Chen et al.(2001) [60]. In the method, one chooses $\frac{1}{N}$ as the fixed threshold.

Then, if the weight of the i th particle is larger than $\frac{1}{N}$, the i th particle will be generated repeatedly $\lfloor Nw_t^i \rfloor$ or $\lfloor Nw_t^i \rfloor + 1$ times. After resampling, the new weights are $\frac{w_t^i}{\lfloor Nw_t^i \rfloor}$ or $\frac{w_t^i}{\lfloor Nw_t^i \rfloor + 1}$. If the weight of the i th particle is less than $\frac{1}{N}$, the i th particle will be sampled with probability Nw_t^i repeatedly. And it is assigned $\frac{1}{N}$ as new weight. The algorithm is given as follow:

Algorithm 10: Reallocation resampling

$$\left[\left\{ \tilde{s}_t^i, \tilde{w}_t^i \right\}_{i=1}^{N^*} \right] = \mathrm{Re}\, allocation \left[\left\{ s_t^i, w_t^i \right\}_{i=1}^{I}, N \right]$$

$i = 0;$

For i=1 to I do

If $w_t^i \geq \frac{1}{N}$

$N_t^i = \lfloor Nw_t^i \rfloor$ (or $N_t^i = \lfloor Nw_t^i \rfloor + 1$)

For $h = 1$ to N_t^i do

$i = i + 1$

$\tilde{s}_t^i = s_t^i$ and $\tilde{w}_t^i = \frac{w_t^i}{N_t^i}$

End

Else

Drawn a random number Δu from a uniform distribution $U(0,\frac{1}{N}]$

If $w_t^i \geq \Delta u$

$i = i + 1$

$\tilde{s}_t^i = s_t^i$ and $\tilde{w}_t^i = \frac{1}{N}$

End

End

End

$N^* = i$

From above algorithm, it is seen that the resampled particles are not equally weighted. Moreover, the sum of all weights is not one, thus in the whole resampling portion, the normalization step is performed as an additional requirement.

In order to reduce the computational complexity, resampling is executed only on some of the particles. First step is to find the particles classified as moderate, negligible, or dominating; then in the second step, various resampling methods are applied on each particle group. There are different resampling functions proposed in [59] for determining which particles are to be resampled or discarded as well as how the weights are allocated.

3.3.3 Regularized Particle Filter

The regularized particle filter (RPF) is developed by Musso, Oudjane et al. (2001) [61]. This is another method to solve the sample impoverishment problem. Most parts of RPF are same with the basic PF algorithm, with the only difference being the resampling stage. PF which resamples form Eq. (11) is a discrete approximation. The RPF resamples from a diffusion kernel function to apply approximation:

$$\hat{f}(s_t|Y_t) \approx \sum_{i=1}^{N_t} w_t^i \cdot K_h(s_t - s_t^i) \tag{20}$$

where the Kernel density function is:

$$K_h(s) = \frac{1}{h^m} K(\frac{s}{h}) \tag{21}$$

From Table 2, m is the dimension of the state vector s, $h > 0$ is the kernel band-width, and w_t^i are normalized weights. The kernel density is a symmetric probability density function satisfying that

$$\int sK(s)ds = 0 \text{ and } \int \|s\|^2 K(s)ds < \infty \tag{22}$$

The kernel and band-width h are chosen to minimize the mean integrated square error (MISE) between the true density and the corresponding regularized empirical representation in Eq. (20). The MISE is

$$MISE = E\left[\int\left[\hat{f}(s_t|Y_t) - f(s_t|Y_t)\right]^2 ds_t\right] \tag{23}$$

If all the samples have the same weight (a special case), the optimal choice of the kernel function is the so-called Epanechnikov kernel [61]

$$K(s)=\begin{cases}\dfrac{n_s+2}{2c_{n_s}}(1-\|s\|^2), & if\ \|s\|<1\\ 0, & otherwise\end{cases} \tag{24}$$

Where c_{n_s} is the volume of the unit hypersphere in $\Re^{n_s}$. In EIS method, the expansion of the family density kernel function will be adopted. The optimal choice of bandwidth h is:

$$h_{opt}=\left[N\left(8\frac{1}{n_s}(n_s+4)\left(2\sqrt{\pi}\right)^{n_s}\right)\right]^{\frac{1}{n_s+4}} \tag{25}$$

The algorithm is given:

Algorithm 11: Regularized resampling

$$\left[\left\{\tilde{s}_t^i, w_t^i\right\}_{i=1}^N\right]=Regularized\left[\left\{s_{t-1}^i, w_{t-1}^i\right\}_{i=1}^N, y_t\right]$$

For i=1 to N do

Draw s_t^i from the know density function $q\left(s\left|s_{t-1}^i, y_t\right.\right)$

Assign the particle weight $w_t^i \propto w_{t-1}^i \dfrac{f(y_t|s_t^i) f(s_t^i|s_{t-1}^i)}{q(s_t^i|s_{t-1}^i, y_t)}$

End for

Normalize weights $w_t^i=\dfrac{w_t^i}{\sum_{j=1}^N w_t^j}$

Compute $\widehat{N_{eff}}=\left(\sum_{i=1}^N\left(w_t^i\right)^2\right)^{-1}$

If $\widehat{N_{eff}}<N$

Compute the empirical covariance matrix M_t of $\left\{s_t^i, w_t^i\right\}_{i=1}^N$

Compute D_t which $D_t D_t^T = M_t$

$$\left[\left\{s_t^i, w_t^i\right\}_{i=1}^N\right] = basicPF\left[\left\{s_{t-1}^i, w_{t-1}^i\right\}_{i=1}^N, y_t\right]$$

For i=1 to N do

Draw ℓ^i from the Epanechnikov Kernel function

$$\tilde{s}_t^i = s_t^i + h_{opt} D_t \ell^i$$

End

End

Even though Eq. (24) and Eq. (25) are for special cases of equally weighted particles and are under Gaussian noise, these methods can still be used in the general case of resampling.

In recent years, some new resampling methods were developed by Li, Sattar et al. (2012, 2014, 2015) [29, 30, 46] to overcome sample degeneracy and impoverishment, with an emphasis on intelligent approaches. They have been investigating methods which are particularly efficient in particle distribution optimization (PDO). PDO is mathematically sound method to alleviate sample degeneracy and impoverishment efficiently, therefore enhanced PF.

When deal with sample degeneracy problem and impoverishment problem, the weight and state can be taken into account for the optimization operation executed. For the PDO method, the state of particles was especially interested to take into account.

4.0 EIS AND ITS APPLICATION

The EIS filter which was developed by Richard and Zhang (2007) [31] is based on auxiliary PF and least-squares regressions. The EIS filter can produce un-biased estimates of state trajectories at high degrees of numerical precision. In particular, the EIS filter can dramatically reduce mean squared errors (MSEs). The EIS filter yields the iterative construction of global approximations of targeted integrands in Eq. (3) and Eq. (4). Here, combining Eq. (3) and Eq. (4), the target density kernel to be approximated is given by

$$\varphi(s_t, s_{t-1}) = f(y_t|s_t)\, f(s_t|s_{t-1})\, \widehat{f}(s_{t-1}|Y_{t-1}) \tag{26}$$

To estimate the integral of Eq. (26) in EIS, the first step is to choose a parametric class kernel function $K = \{k(s_t|s_{t-1}; a_t); a \in A\}$ of auxiliary IS density kernels. Here, the kernel functions are mainly coming from the exponential family of distributions according to the state space model.

$$m_t(s_t|s_{t-1}; a_t) = \frac{k_t(s_t|s_{t-1}; a_t)}{\chi_t(a_t)} \tag{26}$$

For easy notation, in the following part of the thesis, the index of time t will be ignored. And let $\lambda = (s_t, s_{t-1})$, where

$$\chi(a) = \int k(\lambda; a) d\lambda \tag{27}$$

And the integrand can be rewritten as

$$\int \varphi(\lambda) d\lambda = \int \frac{\varphi(\lambda)}{m(\lambda;a)} m(\lambda;a) d\lambda$$
$$= \int \frac{\varphi(\lambda)}{\frac{k(\lambda;a)}{\chi(a)}} m(\lambda;a) d\lambda \tag{28}$$

Similar to the importance sampling, the weight is

$$w(\lambda;a) = \frac{\varphi(\lambda)}{m(\lambda;a)} \tag{29}$$

Draw particles from Eq. (26), then Eq. (28) can be estimated by

$$\int \varphi(\lambda) d\lambda = \chi_1(a_1) \frac{1}{N} \sum_{i=1}^{N} \frac{\varphi(\lambda)}{\frac{k(\lambda;a)}{\chi(a)}} \tag{30}$$

The goal of EIS is to find an optimal $\hat{a} \in A$ which is a vector of parameters. The $\hat{a}$ should minimize the MC variance of the weight over the full support of φ.

$$(\hat{a}, \hat{c}) = \arg\min_{(a,c)} \sum_{i=1}^{N} \left[\ln \varphi(\lambda^i) - c - \ln k(\lambda^i, a) \right]^2 \tag{31}$$

Eq. (31) is a standard least-squares problem. But here, the auxiliary sampling density m itself depends on a. Therefore, a simple example is introduced to explain EIS filter. The example using EIS filter was described in [62]. The State space is given as following:

$$s_t = \delta s_{t-1} + V \eta_t \tag{32a}$$

$$y_t = \beta \exp(\frac{s_t}{2}) \cdot \varepsilon_t \tag{32b}$$

We can also get the density functions according to Eq. (32)

$$f(s_t | s_{t-1}) = \frac{1}{\sqrt{2\pi v}} \exp\left\{ -\frac{(s_t - \delta s_{t-1})^2}{2v^2} \right\} \tag{33a}$$

$$f(y_t|s_t)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left(s_t+\frac{y_t^2}{\beta^2}\exp(-s_t)\right)\right\} \tag{33b}$$

where $\beta=0.6,\delta=0.95,v=0.2,T=1,N=500$. The kernel function also comes from the expansion of family density function:

$$k_t(s_t|s_{t-1};a_t)=\exp\left[-\frac{1}{2}\left(\frac{s_t-u_t}{\sigma_t}\right)^2\right]\cdot\exp\left[-\frac{1}{2}\left(\frac{u_t^2}{\sigma_t^2}-\frac{\delta^2 s_{t-1}^2}{v^2}\right)\right] \tag{34}$$

where

$$\begin{pmatrix}u_t\\ \delta_t^2\end{pmatrix}=\begin{pmatrix}\delta_t^2\left(\frac{\delta s_{t-1}}{v^2}+a_{1t}\right)\\ \frac{v^2}{1-2v^2a_{2t}}\end{pmatrix} \tag{35}$$

The EIS example can be performed by the following steps:

Step 1, call the algorithm 1 to generate the data.

Step 2, give initial value for $t=T$, which is the last time instant. Because in the last period, $\chi_{T+1}=1$ is known, according to Eq. (31), then we can regress $Y=\ln f(y_t|\tilde{s}_t^i)+\ln\chi_{t+1}=a_{0t}+a_{1t}\tilde{s}_t^i+a_{2t}\left(\tilde{s}_t^i\right)^2+u_t^i$ on $X=\begin{bmatrix}1 & \tilde{s}_t^i & \left(\tilde{s}_t^i\right)^2\end{bmatrix}$ to get the optimal estimated parameters $\begin{pmatrix}a_{0t}\\ a_t\end{pmatrix}=\begin{pmatrix}a_{0t}\\ a_{1t}\\ a_{2t}\end{pmatrix}$.

Step 3, track back to find all a_t ; and according to Eq. (35), compute the normal density $m_t(s_t|s_{t-1};a_t)$'s mean and variance: $\begin{pmatrix}u_t\\ \delta_t^2\end{pmatrix}$.

Step 4, we can draw the samples s_t^{i*} from normal distribution $N(u_t,\delta_t^2)$ for each time.

Now we compare the EIS filter and the PF filter (here only use symmetric resampling method) by using the above example to calculate likelihood at time T.

In Table 5, the different mean values of the likelihood values based on the EIS method and the PF method at time T are almost equal. The variance of the likelihood values based on the EIS filter is smaller than that based on PF. Thus we can say that the EIS method is more efficient than the PF method. The EIS filter was applied to Bearings-Only tracking by DeJong et al. (2008) [63]. In [32], the EIS filter offers a significant advance in the empirical analysis of DSGE models. DeJong, Liesenfeld et al. (2015) had tried to use this EIS to estimate real business cycle models [33].

The EIS filter has been applied less than PF because EIS is a new method and its computational complexity is high. But, the EIS filter allows the establishment of optimal continuous IS densities that are global approximations to target integrands. Moreover, the EIS procedure can generate continuous and full adapted samples. These samples can deal with the sample degeneracy problem and impoverishment problem, avoid likelihood discontinuities, and dramatically reduce the MC errors which were produced by likelihood approximations. Some parallel computing algorithms may be used in EIS. And EIS is a more efficient filter and better suited for high-dimension state space models. With these merits, it is expected that in the future, EIS will be applied more in a large number of fields.

Table 5. A comparison of results of PF and EIS

	Likelihood value based on PF	Likelihood value based on EIS
1	0.70694	0.69319
2	0.70911	0.69157
3	0.70307	0.69528
4	0.69957	0.69481
5	0.69629	0.69450
6	0.680579	0.692259
7	0.701422	0.69087
8	0.705163	0.691569
Mean of likelihood value	0.700270	0.693007
Variance of Likelihood value	0.0089	0.001689

5.0 CONCLUSION

KF can estimate linear and Gaussian dynamic models. However, in the most general cases the state space models are nonlinear with non-Gaussian noise which leads to the fast development of nonlinear non-Gaussian filters. In this thesis, the theoretical frameworks of IS, PF, and EIS are reviewed. Specially, the resampling methods of PF to fight sample degeneracy and impoverishment are given in different algorithms of programming. IS is the basic idea to compute the integrals which are required in PF and EIS. For the SIS method, after very few iterations, the sample degeneracy which has only one particle has the particle weight almost close to one, and all other particles' weights are very close to zero will appear. To deal with the problem, many resampling method are introduced, including multinomial resampling, systematic resampling, stratified resampling, residual resampling, residual systematic resampling, branch-kill resampling, rounding-copy resampling, and auxiliary particle filter. However, too much resampling leads to another problem: sample impoverishment, which occurs when few particles have significant weight while most particles with small weight are abandoned during the resampling process. Optimal resampling, rejection control resampling, partial rejection control resampling, reallocation resampling, and regularized particle filter resampling are introduced to fight the problem. Lastly, the EIS filter is reviewed and it is shown that this method is more efficient. In the future, the EIS filter will be applied in more fields, such as signal processing, physics, finance, geography etc.

BIBLIOGRAPHY

1. Smith, G.L., S.F. Schmidt, and L.A. McGee, *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. 1962: National Aeronautics and Space Administration.

2. Julier, S.J. and J.K. Uhlmann. *New extension of the Kalman filter to nonlinear systems*. in *AeroSense'97*. 1997. International Society for Optics and Photonics.

3. Wan, E. and R. Van Der Merwe. *The unscented Kalman filter for nonlinear estimation*. in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. 2000. IEEE.

4. Julier, S.J. and J.K. Uhlmann, *Unscented filtering and nonlinear estimation.* Proceedings of the IEEE, 2004. **92**(3): p. 401-422.

5. György, K., A. Kelemen, and L. Dávid, *Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation.* Procedia Technology, 2014. **12**: p. 65-74.

6. Chen, Z., *Bayesian filtering: From Kalman filters to particle filters, and beyond.* Statistics, 2003. **182**(1): p. 1-69.

7. Gordon, N.J., D.J. Salmond, and A.F. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*. in *IEE Proceedings F (Radar and Signal Processing)*. 1993. IET.

8. Del Moral, P., *Non-linear filtering: interacting particle resolution.* Markov processes and related fields, 1996. **2**(4): p. 555-581.

9. Liu, J.S. and R. Chen, *Blind deconvolution via sequential imputations.* Journal of the american statistical association, 1995. **90**(430): p. 567-576.

10. Kitagawa, G., *Monte Carlo filter and smoother for non-Gaussian nonlinear state space models.* Journal of computational and graphical statistics, 1996. **5**(1): p. 1-25.

11. Isard, M. and A. Blake, *Condensation—conditional density propagation for visual tracking.* International journal of computer vision, 1998. **29**(1): p. 5-28.

12. Fernández-Villaverde, J. and J.F. Rubio-Ramírez, *Estimating macroeconomic models: A likelihood approach.* The Review of Economic Studies, 2007. **74**(4): p. 1059-1087.

13. Lopes, H.F. and R.S. Tsay, *Particle filters and Bayesian inference in financial econometrics.* Journal of Forecasting, 2011. **30**(1): p. 168-209.

14. Flury, T. and N. Shephard, *Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models.* Econometric Theory, 2011. **27**(05): p. 933-956.

15. Kostanjčar, Z., B. Jeren, and J. Cerovec, *Particle filters in decision making problems under uncertainty.* AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 2009. **50**(3-4): p. 245-251.

16. Dore, A., M. Soto, and C.S. Regazzoni, *Bayesian tracking for video analytics.* Signal Processing Magazine, IEEE, 2010. **27**(5): p. 46-55.

17. Soto, A.D.M. and C.S. Regazzoni, *An Overview on Bayesian Tracking for Video Analytics.* 2010.

18. Haykin, S., *Cognitive radio: brain-empowered wireless communications.* Selected Areas in Communications, IEEE Journal on, 2005. **23**(2): p. 201-220.

19. Pardoe, D. and P. Stone. *A particle filter for bid estimation in ad auctions with periodic ranking observations*. in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 2011. International Foundation for Autonomous Agents and Multiagent Systems.

20. Gustafsson, F., et al., *Particle filters for positioning, navigation, and tracking.* Signal Processing, IEEE Transactions on, 2002. **50**(2): p. 425-437.

21. Ristic, B., S. Arulampalam, and N.J. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. 2004: Artech house.

22. Thrun, S. *Particle filters in robotics*. in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. 2002. Morgan Kaufmann Publishers Inc.

23. Gustafsson, F., *Particle filter theory and practice with positioning applications.* Aerospace and Electronic Systems Magazine, IEEE, 2010. **25**(7): p. 53-82.

24. Yardim, C., Z.-H. Michalopoulou, and P. Gerstoft, *An overview of sequential Bayesian filtering in ocean acoustics.* Oceanic Engineering, IEEE Journal of, 2011. **36**(1): p. 71-89.

25. Van Leeuwen, P.J., *Particle filtering in geophysical systems.* Monthly Weather Review, 2009. **137**(12): p. 4089-4114.

26. Doucet, A., N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*, in *Sequential Monte Carlo methods in practice*. 2001, Springer. p. 3-14.

27. Doucet, A., S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering.* Statistics and computing, 2000. **10**(3): p. 197-208.

28. Doucet, A., *On sequential simulation-based methods for Bayesian filtering.* 1998.

29. Li, T., T.P. Sattar, and S. Sun, *Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters.* Signal Processing, 2012. **92**(7): p. 1637-1645.

30. Li, T., et al., *Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches.* Expert Systems with Applications, 2014. **41**(8): p. 3944-3954.

31. Richard, J.-F. and W. Zhang, *Efficient high-dimensional importance sampling.* Journal of Econometrics, 2007. **141**(2): p. 1385-1411.

32. DeJong, D.N., et al., *Efficient Likelihood Evaluation of State-Space Representations.* The Review of Economic Studies, 2012. **80**(2): p. 538-567.

33. DeJong, D.N., et al., *Efficient Likelihood Evaluation of Nonlinear RBC Models.*

34. Daum, F., *Nonlinear filters: beyond the Kalman filter.* Aerospace and Electronic Systems Magazine, IEEE, 2005. **20**(8): p. 57-69.

35. Smith, A., et al., *Sequential Monte Carlo methods in practice*. 2013: Springer Science & Business Media.

36. Del Moral, P., A. Doucet, and A. Jasra, *Sequential monte carlo samplers.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006. **68**(3): p. 411-436.

37. Arulampalam, M.S., et al., *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking.* Signal Processing, IEEE Transactions on, 2002. **50**(2): p. 174-188.

38. Beadle, E.R. and P.M. Djuric, *A fast-weighted Bayesian bootstrap filter for nonlinear model state estimation.* Aerospace and Electronic Systems, IEEE Transactions on, 1997. **33**(1): p. 338-343.

39. Liu, J.S. and R. Chen, *Sequential Monte Carlo methods for dynamic systems.* Journal of the American statistical association, 1998. **93**(443): p. 1032-1044.

40. Carpenter, J., P. Clifford, and P. Fearnhead, *Improved particle filter for nonlinear problems.* IEE Proceedings-Radar, Sonar and Navigation, 1999. **146**(1): p. 2-7.

41. Liu, J.S., R. Chen, and T. Logvinenko, *A Theoretical Framework for Sequential Importance Sampling with Resampling.* 2001: p. 225-246.

42. Bolić, M., P.M. Djurić, and S. Hong, *Resampling algorithms for particle filters: A computational complexity perspective.* EURASIP Journal on Applied Signal Processing, 2004. **2004**: p. 2267-2277.

43. Douc, R. and O. Cappé. *Comparison of resampling schemes for particle filtering*. in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. 2005. IEEE.

44. Hol, J.D., T.B. Schon, and F. Gustafsson. *On resampling algorithms for particle filters.* in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*. 2006. IEEE.

45. Doucet, A. and A.M. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later.* Handbook of Nonlinear Filtering, 2009. **12**: p. 656-704.

46. Li, T., M. Bolic, and P.M. Djuric, *Resampling Methods for Particle Filtering: Classification, implementation, and strategies.* Signal Processing Magazine, IEEE, 2015. **32**(3): p. 70-86.

47. Li, T.-c., et al., *Resampling methods for particle filtering: identical distribution, a new method, and comparable study.* Frontiers of Information Technology & Electronic Engineering, 2015. **16**(11): p. 969-984.

48. Bolić, M., P.M. Djurić, and S. Hong. *New resampling algorithms for particle filters*. in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. 2003. IEEE.

49. Budhiraja, A., L. Chen, and C. Lee, *A survey of numerical methods for nonlinear filtering problems.* Physica D: Nonlinear Phenomena, 2007. **230**(1): p. 27-36.

50. Crisan, D. and T. Lyons, *A particle approximation of the solution of the Kushner–Stratonovitch equation.* Probability Theory and Related Fields, 1999. **115**(4): p. 549-578.

51. Li, T., T.P. Sattar, and D. Tang. *A fast resampling scheme for particle filters*. in *Signal Processing (CIWSP 2013), 2013 Constantinides International Workshop on*. 2013. IET.

52. Pitt, M.K. and N. Shephard, *Filtering via simulation: Auxiliary particle filters.* Journal of the American statistical association, 1999. **94**(446): p. 590-599.

53. Pitt, M.K. and N. Shephard, *Auxiliary variable based particle filters*, in *Sequential Monte Carlo methods in practice*. 2001, Springer. p. 273-293.

54. Whiteley, N. and A.M. Johansen, *Recent developments in auxiliary particle filtering.* Barber, Cemgil, and Chiappa, editors, Inference and Learning in Dynamic Models. Cambridge University Press, 2010. **38**: p. 39-47.

55. Klaas, M., N. De Freitas, and A. Doucet, *Toward practical N2 Monte Carlo: the marginal particle filter.* arXiv preprint arXiv:1207.1396, 2012.

56. Karlsson, R. and N. Bergman. *Auxiliary particle filters for tracking a maneuvering target*. in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*. 2000. IEEE.

57. Kong, A., J.S. Liu, and W.H. Wong, *Sequential imputations and Bayesian missing data problems.* Journal of the American statistical association, 1994. **89**(425): p. 278-288.

58. Fearnhead, P. and P. Clifford, *On‐line inference for hidden Markov models via particle filters.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2003. **65**(4): p. 887-899.

59. Liu, J.S., R. Chen, and W.H. Wong, *Rejection control and sequential importance sampling.* Journal of the American Statistical Association, 1998. **93**(443): p. 1022-1031.

60. Liu, J.S., R. Chen, and T. Logvinenko, *A theoretical framework for sequential importance sampling with resampling*, in *Sequential Monte Carlo methods in practice*. 2001, Springer. p. 225-246.

61. Musso, C., N. Oudjane, and F. Le Gland, *Improving regularised particle filters*, in *Sequential Monte Carlo methods in practice*. 2001, Springer. p. 247-271.

62. Liesenfeld, R. and J.-F. Richard, *Classical and Bayesian analysis of univariate and multivariate stochastic volatility models.* Econometric Reviews, 2006. **25**(2-3): p. 335-360.

63. DeJong, D.N., et al., *Efficient filtering in state-space representations.* Available at SSRN 1337092, 2008.