ENSEMBLE FILTERING FOR STATE SPACE MODELS

BY

JONG HYUN YUN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Professor Yuguo Chen, Chair
Professor John I. Marden
Professor Adam T. Martinsek
Professor Xiaofeng Shao

Abstract

The state space model has been widely used in various fields including economics, finance, bioinformatics, oceanography, and tomography. The goal of the filtering problem is to find the posterior distribution of the hidden state given the current and past observations. The first part of my thesis focuses on designing efficient proposal distributions for particle filters. I propose a new approach named the augmented particle filter (APF), which combines two sets of particles from the observation and state equations. The APF can be applied to general state space models, and it does not require special structures of the model or any approximation to the target or proposal distribution. I find through simulation studies that the APF performs similarly to or better than other filtering algorithms in the literature. The convergence of the augmented particle filter has been established.

The second part of my thesis develops the localization methods for particle filters in high dimensional state space models. Under high dimensional state space models, the computational constraints prevent us from having a large number of particles to avoid the degeneracy problem of the importance weights. When the dimension of the state vector is high, it is common that only a few components of the state vector are dependent on any single component or a set of a few components of the observation vector. In filtering problems, the concept of localization is to use the information in the components of the observation vector to update only the corresponding a few components of the hidden state vector.

I propose the localized augmented particle filter. This new approach divides state vectors into small blocks, and it updates each block of the state vectors through state dynamics and observations. By considering blocks, the influence of observations in updating state vectors is restricted to a few blocks of the state vectors, so the localized augmented particle filter allows constructing the proposal distribution in a lower dimension than the original model. The localized augmented particle filter can outperform many other methods in the literature. The convergence of the localized augmented particle filter has been proved for some class of models.

The method to improve particle filters by dividing the particles into independent batches is presented. The development of the method is motivated by the particle Markov chain Monte Carlo method proposed by Andrieu et al. (2010). Often, the combination of particle filters in batches outperforms the standard particle filter. Parallel computing techniques can be easily adapted to make the implementation fast. The convergence property of the batched particle filter has been established. As the number of batches goes to infinity, the estimate based on the combination of batches converges to the target.

Table of Contents

List of Tables . . . . **vi**

List of Figures . . . . **vii**

Chapter 1 Introduction **1**
1.1 State Space Models 1
1.2 Kalman Filters 2
1.2.1 Ensemble Kalman Filters 3
1.2.2 Localized Ensemble Kalman Filters 4
1.2.3 Serial Updating Ensemble Kalman Filters 4
1.3 Particle Filters 5
1.3.1 Naive Particle Filters 7
1.3.2 Independent Particle Filters 7
1.3.3 Optimal Particle Filters 8
1.4 Resampling Procedures 8
1.4.1 Multinomial Resampling 9
1.4.2 Residual Resampling 9
1.4.3 Stratified Resampling 9
1.5 Curse of Dimensionality 10

Chapter 2 The Augmented Particle Filter **11**
2.1 Augmented Particle Filtering 12
2.2 Justification 16
2.3 State Space Models with Additive Errors 18
2.3.1 SSMs with Gaussian Additive Noise and Linear Observation Equations 19
2.4 Simulation Studies 22
2.4.1 Nonlinear Filtering 22
2.4.2 Maneuvering Target Tacking 24
2.4.3 Linear Gaussian Models 28
2.4.4 Lorenz-96 Model 30

Chapter 3 The Localized Augmented Particle Filter **34**
3.1 The Localized Augmented Particle Filter (LAPF) 34
3.2 The Convergence of the LAPF 35
3.3 Simulation Studies 39
3.3.1 Lorenz-96 Model 39

Chapter 4 Particle Filtering with Independent Batches **42**
4.1 Review of the PF 42
4.2 Independent Batches for the PF 43
4.3 Convergence of the SIRB 45
4.4 Simulation Studies 46

Chapter 5 Future Work **50**
5.1 The Generalized LAPF (GLAPF) 50
5.2 The EnKF as a Proposal Distribution 52
5.2.1 Gaussian Mixture Approximation for the Posterior Density 53

References **55**

List of Tables

2.1 No resampling: The RMSEs and their standard errors of the augmented particle filter (APF), the naive particle filter (NPF), and the independent particle filter (IPF) for different combinations of δ and σ with $T = 5$. 24

2.2 Resampling at every time step: The RMSEs and their standard errors of the augmented particle filter (APF), the naive particle filter (NPF), and the independent particle filter (IPF) for different combinations of δ and σ with $T = 100$. 25

2.3 The RMSEs and their standard errors of three filtering methods for the model in (2.22): the augmented particle filter (APF), the independent particle filter (IPF), and the naive particle filter (NPF). 28

2.4 No Resampling: The RMSEs and their standard errors of three filtering methods for the model in (2.32): the augmented particle filter (APF), the optimal particle filter (OPF), and the ensemble Kalman filter (EnKF). For the APF and the OPF, we provide CV^2 to compare their variances of importance weights. $CV^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(w_T^{(i)}-\bar{w}_T\right)^2}{(\bar{w}_T)^2}$ where $\bar{w}_T = \frac{1}{N}\sum_{i=1}^{N} w_T^{(i)}$. 29

2.5 Resampling at every time step: The RMSEs and their standard errors of four filtering methods for the model in (2.32): the augmented particle filter (APF), the optimal particle filter (OPF), the independent particle filter (IPF), and the naive particle filter (NPF). 29

2.6 Resampling: The RMSEs and their standard errors of four filtering methods for the model in (2.35): the augmented particle filter(APF), the optimal particle filter (OPF), the independent particle filter (IPF), and the naive particle filter (NPF). 33

2.7 No Resampling: The RMSE and their standard errors of four filtering methods for the model in (2.35). the augmented particle filter(APF), the optimal particle filter (OPF), the ensemble Kalman filter (EnKF), and the localized EnKF (LEnKF). $CV^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(w_T^{(i)}-\bar{w}_T\right)^2}{(\bar{w}_T)^2}$ where $\bar{w}_T = \frac{1}{N}\sum_{i=1}^{N} w_T^{(i)}$. 33

3.1 The comparison of RMSE and its standard error for four methods for Lorenz 96 model with $g(x) = x$. 41

3.2 The comparison of RMSE and its standard error for four methods for Lorenz 96 model with $g(x) = \exp(x/||x||^2)$. 41

4.1 A Comparison of the average RMSE and standard errors for the SIR and the SIR with batches based on $K = 100$ repeated experiments. SMC: N=1,000,000, Bat1: N=25 and L=40,000, Bat2: N=100 and L=10,000, Bat3: N=250 and L=4,000, and Bat4: N=1,000 and L=1,000. 48

List of Figures

1.1 A Graphical illustration of the state space model. 1

2.1 The Illustration of the Augmented State Space. 11

2.2 The sparse matrix H used for the data generation in Section 2.4.3. In the gray scale, the brighter component has a larger value than the darker components. 30

3.1 A graphical representation of the covariance matrix Q. Each component of Q is converted to the gray scale with white being 0 and black being 1. 40

4.1 The illustration of the state space model in Chapter 4. 42

4.2 Comparison of the average RMSE for the SIR and the SIR with batches based on $K = 100$ repeated experiments. SMC: N=1,000,000, Bat1: N=25 and L=40,000, Bat2: N=100 and L=10,000, Bat3: N=250 and L=4,000, and Bat4: N=1,000 and L=1,000. 49

Chapter 1

Introduction

1.1 State Space Models

The state space model (SSM) is also called the hidden Markov model. At each time step t, the hidden state x_t evolves through the state equation that describes the first order Markov chain. The x_t's are unobservable vectors, and only a function of x_t with some measurement error is observable through y_t. A graphical illustration of the state space model is given in Figure 1.1. A representation of the state space model is the following:

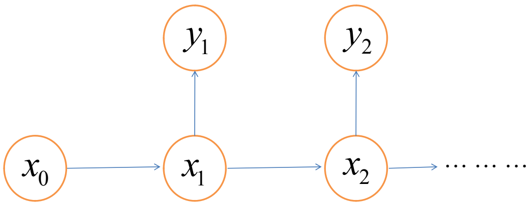

Figure 1.1: A Graphical illustration of the state space model.

$$\begin{cases} y_t|x_t = h_t(x_t, u_t), & \text{the measurement equation,} \\ x_t|x_{t-1} = f_t(x_{t-1}, v_t), & \text{the state equation,} \end{cases} \tag{1.1}$$

where v_t and u_t are independent error terms with known distributions.

The SSM has been widely applied in many fields, including signal processing, image analysis, speech recognition, DNA sequence analysis, oceanography, and time series modeling; see Rabiner (1989), Geweke (1989), Gordon et al. (1995), Elliott et al. (1995), Durbin et al. (1998), Liu and Lawrence (1999), Liu (2001), Tsay (2002), Bertino et al. (2003), and Butala et al. (2009).

One important problem concerning discrete-time SSMs is computing $E(g(X_{0:t})|Y_{1:t})$, the expectation of $g(X_{0:t})$ with respect to the posterior distribution of the hidden state $X_{0:t} = \{X_0, X_1, \ldots, X_t\}$ given the current and past observations $Y_{1:t} = \{Y_1, Y_2, \ldots, Y_t\}$. This is the filtering problem and $E(g(X_{0:t})|Y_{1:t})$ is the

Bayes estimate of $g(X_{0:t})$ with respect to the squared error loss. The filtering problem is typically performed online in the sense that the estimate of $E(g(X_{0:t})|Y_{1:t})$ is needed as soon as the observation y_t arrives. The main focus of this dissertation is the on-line filtering problem.

If we have a linear Gaussian state space model or if the state space is finite, then we can find explicit expressions for the posterior distribution of $X_{0:t}$ given $Y_{1:t} = y_{1:t}$. In the next section, we review the Kalman filter and its variants which can be implemented for the filtering problem when the model is linear Gaussian. In most of other cases, however, $p(x_{0:t}|y_{1:t})$ are not analytically tractable. Thus, we need to pursue a generic method to obtain the estimates in general state space models. The most widely used approach to the filtering problem is the particle filter, which is presented in Section 1.3.

1.2 Kalman Filters

When both the observation and state equations are linear and Gaussian, the Kalman Filter (KF) can be applied to obtain the exact posterior density $p(x_t|y_{1:t})$. The general model setup for Kalman filters is the followings:

$$\begin{cases} y_t|x_t = H_t x_t + u_t, \ u_t \sim N(0, R_t) \\ x_t|x_{t-1} = F_t x_{t-1} + v_t, \ v_t \sim N(0, Q_t). \end{cases} \tag{1.2}$$

In the above equations, F_t and H_t are known matrices. At each time t, the Kalman filter has two steps:

- Analysis step: Compute the mean of variance of $X_t|y_{1:t} \sim N(x_t^a, P_t^a)$, where $x_t^a = x_t^f + K_t(y_t - H_t x_t^f)$, $P_t^a = (I - K_t H_t)P_t^f$, and $K_t = P_t^f H_t'(H_t P_t^f H_t' + R_t)^{-1}$ (the Kalman gain matrix).
- Forecast step: Compute the mean of variance of $X_{t+1}|y_{1:t} \sim N(x_{t+1}^f, P_{t+1}^f)$, where $x_{t+1}^f = F_t x_t^a$ and $P_{t+1}^f = F_t P_t^a F_t' + Q_t$.

The Kalman filter is applicable only if we have linear Gaussian state space models. The extended Kalman filter (EKF) is designed to extend the applicability of Kalman filtering by linearizing nonlinear functions. Here, we allow both equations to be nonlinear, but the distributions for the error terms are still normal:

$$\begin{cases} y_t|x_t = h_t(x_t, u_t), \ u_t \sim N(0, R_t) \\ x_t|x_{t-1} = f_t(x_{t-1}, v_t), \ v_t \sim N(0, Q_t). \end{cases}$$

Let $H_t = Dh_t(\cdot)|_{(x_t^f,0)}$ and $F_t = Df_t(\cdot)|_{(x_t^a,0)}$ where $Df(\cdot)|_x$ denotes a Jacobian matrix of function $f(\cdot)$ at x. The KF is implemented as follows on the linearized system:

- Analysis step: Compute $x_t^a = x_t^f + K_t(y_t - H_t x_t^f)$, $P_t^a = (I - K_t H_t)P_t^f$, and $K_t = P_t^f H_t'(H_t P_t^f H_t' + R_t)^{-1}$, where $x_t^f = f_t(x_{t-1}^a, v_t)$.

- Forecast step: Compute $x_{t+1}^f = F_t x_t^a$ and $P_{t+1}^f = F_t P_t^a F_t' + Q_t$, where $x_t^a = h_t(x_t^a, 0)$.

The EKF in general does not give consistent estimates of the state. In this approach, it is crucial to have an accurate linear approximation of the nonlinear functions. If the nonlinearities are very severe, then the solutions from the EKF would be far from the true state.

1.2.1 Ensemble Kalman Filters

When both the state and measurement equations are linear and Gaussian, we can implement the Kalman filtering which gives analytic solutions for $E(X_t|Y_{1:t})$ and $E(X_{t+1}|Y_{1:t})$. When the dimension of the state vector x_t is high, it is hard to apply the Kalman filter for two reasons: First, the computation of matrix product for large covariance matrices takes a large amount of CPU time. Second storing the large covariance matrix takes a lot of space. For example, if the state vector x_t is represented by 128^3 components, then 8 TB of storage is required to store the matrix P_t^f with 32-bit precision. Thus, we want to pursue a filter with lower computational cost. The following is the ensemble Kalman filter (EnKF) updating algorithm proposed by Evensen (1994), which can overcome the computation and storage problems:

1. Compute $x_t^{a,(i)} = x_t^{f,(i)} + \hat{K}_t(y_t + v_t^{(i)} - H_t x_t^{f,(i)})$ where $\hat{K}_t = \hat{P}_t^f H_t'(H_t \hat{P}_t^f H_t' + R_t)^{-1}$ and $v_t^{(i)} \sim N(0, R_t)$.

2. Estimate x_t^a and P_t^a by these ensembles (or particles):

$$\hat{x}_t^a = \frac{1}{N}\sum\nolimits_{i=1}^N x_t^{a,(i)}, \ \hat{P}_t^a = \frac{1}{N}\sum\nolimits_{i=1}^N (x_t^{a,(i)} - \hat{x}_t^a)(x_t^{a,(i)} - \hat{x}_t^a)'.$$

3. For the next time step $t+1$, each ensemble is propagated by $x_{t+1}^{f,(i)} = F_t x_t^{a,(i)} + u_t^{(i)}$, $u_t^{(i)} \sim N(0, Q_t)$.

4. In the same way, x_{t+1}^f and P_{t+1}^f can be estimated from $x_{t+1}^{f,(i)}$'s as follows:

$$\hat{x}_{t+1}^f = \frac{1}{N}\sum\nolimits_{i=1}^N x_{t+1}^{f,(i)}, \ \hat{P}_{t+1}^f = \frac{1}{N}\sum\nolimits_{i=1}^N (x_{t+1}^{f,(i)} - \hat{x}_{t+1}^f)(x_{t+1}^{f,(i)} - \hat{x}_{t+1}^f)'. \quad (1.3)$$

Note that we do not really need to calculate or store $\hat{P}_t^a$, since the ensembles can be updated without $\hat{P}_t^a$. Also, the estimate from the EnKF converges to the solution from the Kalman filter as N goes to infinity (Butala et al., 2008).

1.2.2 Localized Ensemble Kalman Filters

When the dimension of the state vector x_t is high, it is common that only a few components of x_t are dependent on any single component or a set of a few components of y_t. In filtering problems, the concept of localization is to use the information in the components of y_t to update only the corresponding a few components of x_t. In the EnKF, it can be done by introducing the constrained covariance matrix estimator $C \circ \hat{P}_t^f$, where $\circ$ denotes component–wise matrix product (also called the Schur product). A covariance tapering matrix C can be chosen from our prior knowledge about the state space model. For example, the (i, j)-th component of C can be chosen as a distance measurement between the i-th and j-th components of x_t. The most popular choice of C is the banded matrix with a few bands around the diagonal components.

The localized ensemble Kalman filter (LEnKF) provides a much more stable estimate in high dimensional state space model when it is applied with a good choice of C (Furrer and Bengtsson, 2007). One interpretation is because the localization regularizes the estimate of the covariance matrix (Bickel and Levina, 2008), so we can obtain a stable covariance estimate with a small bias when we can only afford a few samples. Note that the localized ensemble Kalman filter introduces a bias in our estimate, and it has been shown that the LEnKF converges to the local Kalman filter (Butala et al., 2009). The EnKF or the LEnKF can be implemented even for nonlinear or non–Gaussian state space models. However, the convergence properties of the estimates cannot hold anymore.

1.2.3 Serial Updating Ensemble Kalman Filters

In Step 1 of the EnKF algorithm, the inversion of a large covariance matrix when the dimension of y_t is high, is another computationally expensive step, and the result may be unstable. Sometimes in the state space model, the components of y_t in (1.2) are conditionally independent given x_t. In this case, the observation y_t can be processed one at a time, and x_t can be updated by assimilating the observation y_t serially. We can utilize this property to reduce the computation cost of matrices inversion. Let $y_{t,j}$ denote the j-th block of the conditionally independent components for $j = 1, ..., M$. Then, we can implement the following to achieve the EnKF estimate at each time t:

1. For each $j = 1, \ldots, M$, find the sub–matrices of H_t and R_t which correspond to $y_{t,j}$, and denote them by $H_{t,j}$ and $R_{t,j}$, respectively.

2. When $j = 1$, we have $x_{t,1}^{a,(i)} = x_t^{f,(i)} + \hat{K}_{t,1}(y_{t,1} + v_{t,1}^{(i)} - H_{t,1}x_t^{f,(i)})$, $v_{t,1}^{(i)} \sim N(0, R_{t,1})$ where $\hat{K}_{t,1} = \hat{P}_t^f H_{t,1}'(H_{t,1}\ \hat{P}_t^f H_{t,1}' + R_{t,1})^{-1}$.

3. Estimate the mean and covariance as follows:

$$\hat{x}^a_{t,j} = \frac{1}{N}\sum\nolimits_{i=1}^{N} x^{a,(i)}_{t,j},\ \hat{P}^a_{t,j} = \frac{1}{N}\sum\nolimits_{i=1}^{N} (x^{a,(i)}_{t,j} - \hat{x}^a_{t,j})(x^{a,(i)}_{t,j} - \hat{x}^a_{t,j})'.$$

4. For $j > 1$, update $x^{a,(i)}_{t,j} = x^{a,(i)}_{t,j-1} + \hat{K}_{t,j}(y_{t,j} + v^{(i)}_{t,j} - H_{t,j}x^{a,(i)}_{t,j-1})$, $v^{(i)}_{t,j} \sim N(0, R_{t,j})$ where $\hat{K}_{t,j} = \hat{P}^a_{t,j-1}H'_{t,j}(H_{t,j}\hat{P}^a_{t,j-1}H'_{t,j} + R_{t,j})^{-1}$.

5. Repeat Steps 3 and 4 until all M blocks are updated.

6. For the next time step $t+1$, each ensemble is propagated by $x^{f,(i)}_{t+1} = F_t x^{a,(i)}_{t,M} + u^{(i)}_t$, $u^{(i)}_t \sim N(0, Q_t)$, and then obtain $\hat{P}^f_{t+1}$.

We can see that for each $y_{t,j}$ updating $x^a_{t,j}$ is computationally simple because we deal with lower dimensional matrix inversion problems with serial updating.

1.3 Particle Filters

The particle filter (PF), also known as sequential importance sampling (SIS), is a method to generate a weighted sample $\{x^{(i)}_{0:t}, w^{(i)}_t\}_{i=1}^N$ from the posterior distribution of $X_{0:t}$ given $Y_{1:t} = y_{1:t}$. Based on the weighted sample, we are able to estimate $E(g(X_{0:t})|Y_{1:t})$ at each t. Also, due to the size of the state vector x_t and the time step t, the dimension of $X_{0:t}$ does not allow us to draw particles all at once. To deal with this issue, the PF considers a recursive way to generate weighted samples as follows:

The posterior density can be decomposed into the product of each state and measurement density.

$$\begin{aligned} p(x_{0:t}|y_{1:t}) &\propto p(y_t|x_t)p(x_t|x_{t-1})p(x_{1:t-1}|y_{1:t-1}) \\ &\propto p(x_0)\prod_{n=1}^{t} p(x_n|x_{n-1})p(y_n|x_n). \end{aligned}$$

Assuming $p(x_0)$ is known, $E(g(x_{0:t})|y_{1:t})$ can be rewritten as

$$\int\cdots\int g(x_{0:t})C_t\frac{p(x_0)\prod_{n=1}^{t} p(x_n|x_{n-1})p(y_n|x_n)}{p(x_0)\prod_{n=1}^{t} q(x_n|y_n, x_{n-1})} \\ \times p(x_0)\prod_{n=1}^{t} q(x_n|y_n, x_{n-1})dx_0\cdots dx_t,$$

where C_t is a normalizing constant coming from the unnormalized densities in the integrand. To draw particles, we draw samples from a proposal density $q(x_n|y_n, x_{n-1})$ for each n. By doing the above decomposition, we are able to draw the weighted samples in the recursive way as follows:

1. Draw $x_0^{(i)}$ from $p(x_0)$ for $i = 1, \dots, N$.

2. For each time-step $t = 1, \dots, T$, draw $x_t^{(i)}$ from the proposal distribution $q(x_t|y_t, x_{t-1}^{(i)})$. Compute the importance weight as
$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})}, \tag{1.4}$$
and normalize the weight by $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}$.

Note that in practice we often only know the densities in (1.4) up to some normalizing constants. Using the normalized importance weight $\tilde{w}_t^{(i)}$ in Step 2 guarantees the convergence without computing the normalizing constants in $w_t^{(i)}$. From the theory of the general simple importance sampling, we have the convergence of our estimate at time t as follows:

$$\sum_{i=1}^N \tilde{w}_t^{(i)} g(X_{0:t}^{(i)}) \xrightarrow{p} E(g(X_{0:t})|Y_{1:t}) \text{ as } N \to \infty.$$

More convergence results can be found in Doucet et al. (2000).

The performance of the particle filter depends highly on the quality of the proposal density. One way to measure the quality is to look at the variance of the importance weights. The best scenario is to draw samples from $p(x_{0:t}|y_{1:t})$, so all the importance weights are the same. In this case, the variance of the weight is 0. However, it is usually impossible to sample from $p(x_{0:t}|y_{1:t})$ except some special SSMs. If the proposal density is far from $p(x_{0:t}|y_{1:t})$, the variance of the weight w_t would be large. In the worst case scenario, only one particle with the largest weight would dominate our estimate, which is called the degeneracy of the weights. See Bengtsson et al. (2008) for more details about the sample size requirements to avoid the degeneracy problem. The variance of the weight is also used in the effective sample size N_{ess} introduced in Kong et al. (1997) which is:
$$N_{ess} = \frac{N}{1 + Var(w_t)} = \frac{N}{E(w_t^2)} \leq N. \tag{1.5}$$

We estimate the effective sample size by its sample counterpart as follows:

$$\hat{N}_{ess} = \frac{1}{\sum_{i=1}^N (\tilde{w}_t^{(i)})^2}.$$

Note that $\hat{N}_{ess}$ must lie between 1 and N. A common problem in particle filtering is that as t increases, the variance of the weight increases, so N_{ess} gets close to 1, which implies that applying the PF alone could be very ineffective. We will consider this issue in the last section of this chapter. In the next a few sections, we can see filtering algorithms based on different proposal densities.

1.3.1 Naive Particle Filters

The naive particle filter (NPF) is proposed by Gordon et al. (1993). It chooses the proposal as

$$q(x_t|y_t, x_{t-1}) = p(x_t|x_{t-1}).$$

Also, the weight evaluation is quite simple for this case:

$$w_t = w_{t-1}p(y_t|x_t).$$

Hence, no information from the observation y_t is used to generate the particles at time t. When combining the information from both equations is too challenging, or when drawing the particles from the state equation is doable, but computing the state density is impossible, we can implement the naive particle filter. Since the information in y_t is incorporated only by the importance weights, this approach does not work well in the state space model with accurate measurement equations and noninformative state equations. In the opposite case with noisy measurement and accurate state equations, we expect the performance of the NPF to be fine.

1.3.2 Independent Particle Filters

The independent particle filter (IPF), introduced by Lin et al. (2005), is another extreme case because the construction of the proposal density is completely based on the measurement equation

$$q(x_t|y_t, x_{t-1}) = q(x_t|y_t).$$

Note that the particles from the past do not appear in the proposal density. Only when $p(y_t|x_t)$ is integrable with respect to x_t, we can choose $q(x_t|y_t) \propto p(y_t|x_t)$; otherwise we choose $q(x_t|y_t)$ to be close to $p(y_t|x_t)$.

Since the particles generated from the proposal are independent of the past particles, the particles at time t can be matched with any particles in the history by a random permutation. After the matching is done, we can calculate the importance weight as

$$w_t = w_{t-1}\frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|y_t)}.$$

Also, we can consider multiple matchings from several random permutations. Then, the weight will be a simple average of the weights from each matching. This strategy can reduce the variance of the weights (Lin et al., 2005). However, the resampling procedure in the IPF does not help to control the variance of the estimate effectively.

1.3.3 Optimal Particle Filters

The optimal particle filter (OPF) can be applied when the following proposal density is available:

$$q(x_t|y_t, x_{t-1}) = p(x_t|y_t, x_{t-1}).$$

The reason we call it optimal is because this proposal density minimizes the variance of the weights among other choices of the proposals (Doucet and Gordon, 1999).

However, the OPF can be implemented for only very limited class of SSMs, because the proposal and the weight update are intractable in general. One class of models that the optimal particle filter can be implemented is

$$\begin{cases} y_t|x_t = H_t x_t + u_t,\ u_t \sim N(0, R_t) \\ x_t|x_{t-1} = f_t(x_{t-1}) + v_t,\ v_t \sim N(0, Q_t), \end{cases} \tag{1.6}$$

where f_t is any function and H_t is a matrix. With the above model, the proposal density for the optimal particle filter is $N(\mu_t, \Sigma_t)$, where

$$\Sigma_t = (Q_t^{-1} + H_t' R_t^{-1} H_t)^{-1} \text{ and } \mu_t = \Sigma_t (Q_t^{-1} f(x_{t-1}) + H_t' R_t^{-1} y_t).$$

Also, the weight evaluation can be written as $w_t = w_{t-1} p(y_t|x_{t-1})$, where $p(y_t|x_{t-1})$ is

$$N(y_t|H_t f(x_{t-1}), R_t + H_t' Q_t H_t).$$

1.4 Resampling Procedures

One source of high dimensionality in particle filtering is the time step t. If we perform the PF for large t, the target density $p(x_{1:t}|y_{1:t})$ would become a high dimensional density even with the low dimensional x_t. Notice that the variance of the importance weight increases over time, so one can expect that as t goes to infinity, our estimate of $E(X_{0:t}|Y_{1:t})$ will be evaluated by a single particle because the maximum of $\tilde{w}_t^{(i)}$ goes to 1 (Bengtsson et al., 2008).

Resampling is proposed to overcome the degeneracy of the weight. In the simple random resampling procedure, the given particles will be resampled with probability proportional to their importance weights.

In other words, let N_i denote the number of times the i-th particle $x_{1:t}^{(i)}$ appears after resampling. Then, we have $E(N_i) = N\tilde{w}_t^{(i)}$, which implies that we put more effort on the particle with larger weight to be evolved over time. After the resampling is done, we set $\tilde{w}_t^{(i)} = 1/N$ for each sample. If we implement the resampling procedure at every time t, then the convergence result of the particle filtering still holds with resampling. The variance of the filtering estimate under resampling can be found in Doucet and Johansen (2011).

In this section, we review several resampling procedures that have been introduced to achieve smaller $Var(N_i)$, the additional source of variation introduced by the resampling, or to reduce the computational cost of the resampling procedure.

1.4.1 Multinomial Resampling

The multinomial resampling is also known as the bootstrap resampling. Let $\mathbf{N} := (N_1, \ldots, N_N)$ and $\tilde{\mathbf{w}}_\mathbf{t} := (\tilde{w}_t^{(1)}, \ldots, \tilde{w}_t^{(N)})$. Then, we sample $\mathbf{N}$ from $multinomial(N; \tilde{\mathbf{w}}_\mathbf{t})$.

1.4.2 Residual Resampling

The residual resampling is also know as the remainder resampling. Here N_i can be obtain as follows:

$$N_i = \lfloor \tilde{w}_t^{(i)} N \rfloor - \bar{N}_i, \;\; i = 1, \ldots, N$$

where $\lfloor \cdot \rfloor$ denote the integer part and $(\bar{N}_1, \ldots, \bar{N}_N)$ is a sample from $multinomial(N - R; \tilde{\mathbf{w}}_\mathbf{t})$ with $R := \sum_{i=1}^N \lfloor \tilde{w}_t^{(i)} N \rfloor$. In this approach, R particles are selected deterministically given $\tilde{\mathbf{w}}_\mathbf{t}$. Thus, the residual resampling would be effective to obtain small $Var(N_i)$.

1.4.3 Stratified Resampling

First, we partition an interval $(0, 1]$ into N disjoint intervals whose length equal to $\tilde{\mathbf{w}}_\mathbf{t}$, respectively. Then, we draw $U_j \sim U((\{j-1\}/N, j/N])$ for $j = 1, \ldots, N$, and N_i is chosen by counting the number of U_j's in the i-th pre–partitioned interval. This approach is faster than the other methods, so we implement the stratified resampling for the simulation studies in the following chapters. See Douc (2005) for more details about properties of each resampling schemes.

1.5 Curse of Dimensionality

There are two sources of high dimensionality in particle filtering. One is the dimension of the state vector x_t, and the other is the time step t. The resampling procedure can deal with problems caused by large t in the particle filter. We need to find an effective approach to handle a high dimensional x_t because the degeneracy of the importance weights can result from the high dimensionality of x_t alone in the filtering problem.

The reasons that the EnKF works for the inference in high dimensional state space models are: 1) it deals with the posterior density $p(x_t|y_{1:t})$ instead of $p(x_{0:t}|y_{1:t})$, 2) in linear Gaussian models, the EnKF converges to the optimal estimate for $E(X_t|Y_{1:t})$, and 3) it allows the localization by taking into account the dependency relation between the observation and the state vectors. Roughly speaking, the LEnKF gives a shrinkage estimate by restricting the influence of the observation y_t on only some components of x_t.

Notice that the LEnKF only works only for the linear Gaussian models. We want to construct a particle filtering method working in general nonlinear non–Gaussian models. Applying the localization idea to particle filtering is not straightforward, and it will be investigated in the following chapters.

This dissertation is structured as follows. Chapter 2 describes the development of a new particle filter algorithm called the augmented particle filter. Chapter 3 describes the localization procedure that can be implemented with the augmented particle filter. Chapter 4 describes an effective way to combine particle filtering estimates coming from independent identical batches of particle filtering. Chapter 5 describes some future work on the particle filter in high dimensional state space models.

Chapter 2

The Augmented Particle Filter

In this chapter, we introduce a new particle filter named the augmented particle filter (APF). Our framework is not restricted to Sequential Monte Carlo (SMC) or state space models (SSMs). However, we will show the development of APFs under the SSM framework. APFs depend on the combination of observation and state equations to construct a proposal distribution. The implementation of APFs does not require special structures of SSMs or any approximation to the target or proposal distribution which may affect the convergence of our estimates. To be more specific, APFs combine two sets of particles from the observation equation (likelihood function) and the state equation (prior distribution). To avoid the difficulty in the evaluation of importance weights, we augment the state space and specify the joint proposal distribution. We find that the augmented state space does not hurt the efficiency of the filtering algorithms, and often times the APF performs better than other filtering algorithms in the literature.

Here, we introduce the SSM with the augmented state space. In Figure 2.1, a few more nodes with the superscript f are added the SSM to illustrate the idea. Given x_{t-1}, the augmented state vector x_t^f depends only on x_{t-1} through the state equation $p(x_t^f|x_{t-1})$, and x_t^f is free of all the observations, other state vectors, and other augmented state vectors except x_{t-1}. The augmented state space model can be redefined as (2.1).

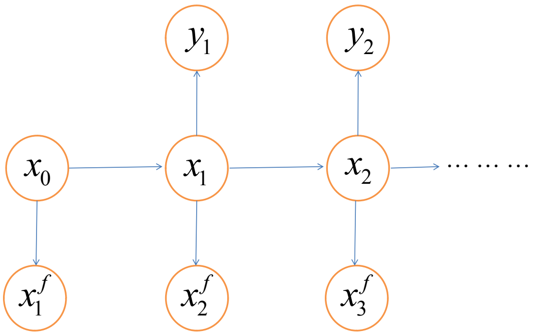

Figure 2.1: The Illustration of the Augmented State Space.

$$\begin{cases} y_t|x_t = h_t(x_t, u_t), & \text{the observation equation,} \\ x_t|x_{t-1} = f_t(x_{t-1}, v_t), & \text{the state equation,} \\ x_t^f|x_{t-1} = f_t(x_{t-1}, v_t^f), & \text{the augmented state equation,} \end{cases} \tag{2.1}$$

where the error term v_t^f can be defined by users, and as a default choice we can set $v_t^f \overset{d}{=} v_t$.

By considering the augmented state space, we actually make the state space two times larger than the original state space, since now our target distribution is $p(x_{0:t}, x_{1:t}^f|y_{1:t})$ instead of $p(x_{0:t}|y_{1:t})$. However, we can sample from the augmented state space $x_{1:t}^f$ directly. Notice that the convergence of SMC estimates does not rely on the dimension of the state space when we directly draw particles from the target distribution. For each augmented state vector x_t^f, we have its target distribution as $p(x_t^f|x_{t-1})$, from which we can sample directly. In following sections, we will see how the APFs can utilize this new structure of SSMs.

2.1 Augmented Particle Filtering

For general state space models, the NPF is often used because the OPF is usually not available, and incorporating the observation y_t into the proposal density could be too challenging. For the NPF, the proposal density may not be close to the target density which could lead to a large variance of the importance weight. If the dimension of the model is high, the problem would be even worse. The proposal density of the NPF relies solely on the state equation, so it does not use any information from the observation y_t. The proposal density of the APF combines the information from both the observation and state equations, which makes it possible for the APF to outperform the NPF.

In this section, we propose the APF algorithm for general SSMs given in (1.1). In such SSMs, we evaluate the amount of information contained in the two equations by looking at conditional variances of $y_t|x_t$ and $x_t|x_{t-1}$. Thus, at each time t, the implementation of the APF requires evaluating $Var(y_t|x_t)$ and $Var(x_t|x_{t-1})$ to decide the weight to put on observation and state equations, so we can build up the proposal distribution as a combination of the two equations. However, state vectors are unknown, so we have to estimate the variance terms by linearizing the equations and plugging in PF estimates into the equations unless the model has additive errors. Before we describe the algorithm, we explain a few notations: $x_t^{(i)}$ and $x_t^{f,(i)}$ denote samples generated for the hidden state vectors shown in Figure 2.1, and $x_t^{l,(i)}$ denotes a sample from a proposal distribution which solely depends on the likelihood function associated with the current observation y_t.

The detailed algorithm of the APF for the above state space model is given as follows. At the initial step

$t = 0$, draw $x_0^{(i)}$ from $q(x_0)$, whose target density is $p(x_0)$, for $i = 1, \dots, N$, and compute the importance weight as

$$w_0^{(i)} = \frac{p(x_0^{(i)})}{q(x_0^{(i)})}.$$

For $i = 1, \dots, T$, we repeat the following steps:

1. Draw $x_t^{f,(i)}$ from $p(x_t^f | x_{t-1}^{(i)})$, which can be easily obtained by evolving through the augmented state equation.

2. Draw $x_t^{l,(i)}$ from a proposal density $q_l(x_t^l | y_t)$ whose functional form is close to $p(y_t | x_t)$.

3. Let $\tilde{h}_t(\tilde{x}_t, u_t)$ denote the derivative of $h_t(x_t, u_t)$ with respect to x_t at $\tilde{x}_t$ where $\tilde{x}_t$ is an temporary estimate of x_t. Evaluate $\hat{H}_t := E(\tilde{h}_t(\tilde{x}_t, u_t)|\tilde{x}_t)$ and $\hat{R}_t := Var(h_t(\tilde{x}_t, u_t)|\tilde{x}_t)$.

4. Evaluate $\hat{Q}_t := Var(f(\bar{x}_{t-1}, v_t^f)|\bar{x}_{t-1})$.

5. Let $\hat{\Sigma}_t = (\hat{H}_t' \hat{R}_t^{-1} \hat{H}_t)^{-1}$. Then, combine the two particles from Steps 1 and 2 as

$$\begin{aligned} x_t^{(i)} &= (\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1} (\hat{\Sigma}_t^{-1} x_t^{l,(i)} + \hat{Q}_t^{-1} x_t^{f,(i)}) \\ &= \hat{Q}_t (\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{l,(i)} + \hat{\Sigma}_t (\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{f,(i)}. \end{aligned} \tag{2.2}$$

6. Calculate the importance weight of $x_t^{(i)}$ as

$$w_t^{(i)} = \frac{p(y_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)})}{q_l(x_t^{l,(i)} | y_t)} w_{t-1}^{(i)}.$$

The construction of the APF proposal can be viewed as follows: First, we draw a forecast particle $x_t^{f,(i)} = f_t(x_{t-1}^{(i)}, v_t^{f,(i)})$ by evolving the particle according to the state equation, and draw a likelihood particle $x_t^{l,(i)}$, which comes from a proposal density for the likelihood function $p(y_t|x_t)$. Then, we combine the two sets of particles to incorporate the information contained in both the observation and state equations. The justification of the weight computation in Step 6 will be given in Section 2.2.

Here are a few remarks on the general APF algorithm:

1. Note that to choose $q_l(x_t^l|y_t)$ proportional to the observation density, $p(y_t|x_t)$ must be proper with respect to x_t, in such case we prefer to choose $q_l(x_t^l|y_t) \propto p(y_t|x_t^l)$. As an alternative, when the SSM has the additive observation noise, we can linearize $h(x_t)$ w.r.t x_t at the modes of the likelihood $p(y_t|x_t)$, and then we can substitute $h(x_t)$ with its linearization in $p(y_t|x_t)$ to construct $q_l(x_t^l|y_t)$.

2. In the case that the function $h_t(x_t, u_t)$ is not differentiable at $\tilde{x}_t$ or if it is impossible to obtain the analytical derivative of $h_t(x_t, u_t)$, we could set $\tilde{h}_t(x_t, u_t)$ as a numerical differentiation at $(\tilde{x}_t, E(u_t))$.

3. When coefficients in the linear combination $\hat{H}_t$, $\hat{R}_t$, or $\hat{Q}_t$ in (2.2) cannot be computed analytically, we can use Monte Carlo method to estimate. Let $u_t^{(i)}$ and $v_t^{(i)}$ denote samples from their own error distributions, we have

$$\begin{aligned} \hat{H}_t &\approx \frac{1}{N}\sum\nolimits_{i=1}^{N} \tilde{h}_t(\tilde{x}_t, u_t^{(i)}) \\ \hat{R}_t &\approx \frac{1}{N}\sum\nolimits_{i=1}^{N} (h_t(\tilde{x}_t, u_t^{(i)}) - h_t(\tilde{x}_t, E(u_t)))(h_t(\tilde{x}_t, u_t^{(i)}) - h_t(\tilde{x}_t, E(u_t)))' \\ \hat{Q}_t &\approx \frac{1}{N}\sum\nolimits_{i=1}^{N} (f(\bar{x}_{t-1}, v_t^{(i)}) - f(\bar{x}_{t-1}, E(v_t)))(f(\bar{x}_{t-1}, v_t^{(i)}) - f(\bar{x}_{t-1}, E(v_t)))'. \end{aligned} \tag{2.3}$$

In Section 2.2, we will see that the coefficient estimation above would not cause any problems in the convergence of the APF estimates.

4. When $p(y_t|x_t)$ is a proper density w.r.t. x_t, we can take an optional resampling step right after Step 2. The importance weight at step 2 can be computed as

$$w_t^{l,(i)} = \frac{p(y_t|x_t^{l,(i)})}{q_l(x_t^{l,(i)}|y_t)}. \tag{2.4}$$

We resample $x_t^{l,(i)}$ with probability proportional to $w_t^{l,(i)}$, so $x_t^{l,(i)}$ follows $p(x_t^l|y_t)$ approximately. If $p(y_t|x_t)$ is not a proper density in terms of x_t, then this step is not possible because $w_t^{l,(i)}$ would not be the proper weight.

5. We usually choose the temporary estimate $\tilde{x}_t$ as the mode of $p(y_t|x_t)$ (viewed as a function of x_t with y_t fixed). If it is difficult to find the mode, we can obtain the temporary estimate by generating samples from $q_l(x_t^l|y_t)$. Then, we have

$$\tilde{x}_t = \frac{\sum_{j=1}^{N} w_t^{l,(j)} x_t^{l,(j)}}{\sum_{j=1}^{N} w_t^{l,(j)}},$$

which is the estimate of $E(X_t|Y_t)$ with the flat prior of x_t.

6. Besides the linear combination in (2.2), the APF allows other ways to combine the forecast and likelihood particles. For example, if we believe the state equation is not informative, we can inflate the variance components in the augmented state noise v_t^f, so the final particle would put more weight on $x_t^{l,(i)}$ which is the particle from the IPF proposal. This is one extreme case. Another extreme case is to inflate the variance components in $\hat{\Sigma}_t$ when the observation equation is not informative. In this

case, the final particle would be close to $x_t^{f,(i)}$, which is the particle from the NPF proposal. Thus, the NPF and IPF can be viewed as two extreme cases of the APF.

7. The particle $x_t^{l,(i)}$ and $x_t^{f,(i)}$ can be matched arbitrarily. The APF can adopt the multiple matching technique proposed by Lin et al. (2005) to reduce the variance of the importance weight. Let $K_m = k_{m,1}, \ldots, k_{m,N}$ denote the set of random permutation of $(1, \ldots, N)$. For $m = 1, \ldots, M$, each $x_t^{l,(i)}$ can be combined with the permuted forecast particle through the linear combination

$$x_{t,m}^{(i)} = \hat{Q}_t(\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{l,(i)} + \hat{\Sigma}_t(\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{f,(k_{m,i})}, \tag{2.5}$$

and compute the importance weight as

$$w_{t,m}^{(i)} = \frac{p(y_t|x_{t,m}^{(i)})p(x_{t,m}^{(i)}|x_{t-1}^{(k_{m,i})})}{q_l(x_t^{l,(i)}|y_t)} w_t^{(k_{m,i})}. \tag{2.6}$$

After obtaining $M \times N$ weighted samples $\{x_{t,m}^{(i)}, w_{t,m}^{(i)}, i = 1, \ldots, N, m = 1, \ldots, M\}$, we estimate $E(g(X_t)|Y_{1:t})$ by

$$\hat{E}(g(X_t)|Y_{1:t}) = \frac{\sum_{m=1}^M \sum_{i=1}^N w_{t,m}^{(i)} \cdot g(x_{t,m}^{(i)})}{\sum_{m=1}^M \sum_{i=1}^N w_{t,m}^{(i)}}.$$

To obtain N particles to evolve to the next time step $t+1$, we perform the selection step as follows. For each i we select one particle from $\{x_{t,m}^{(i)}\}_{m=1}^M$ with probabilities proportional to $\{w_{t,m}^{(i)}\}_{m=1}^M$, and set the selected particle and its history to be $x_{0:t}^{(i)}$ with the importance weight

$$w_t^{(i)} = \frac{\sum_{m=1}^M w_{t,m}^{(i)}}{M}. \tag{2.7}$$

The APF proposal draws $x_t^{f,(i)}$ in the first step and constructs the final particle $x_t^{(i)}$ by combining $x_t^{f,(i)}$ and $x_t^{l,(i)}$. Thus, the APF proposal can be viewed as a joint distribution:

$$q(x_t^{(i)}, x_t^{f,(i)}|y_t, x_{t-1}^{(i)}) = q(x_t^{(i)}|x_t^{f,(i)}, y_t)p(x_t^{f,(i)}|x_{t-1}^{(i)}).$$

From (2.11), we have $q(x_t^{(i)}|y_t, x_t^{f,(i)}) = q_l(x_t^{l,(i)}|y_t)|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|$, so the marginalized proposal density $q(x_t^{(i)}|x_{t-1}, y_t)$ can be evaluated through

$$q(x_t^{(i)}|y_t, x_{t-1}^{(i)}) = \int q_l(\hat{\Sigma}_t\{(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})x_t^{(i)} - \hat{Q}_t^{-1}x_t^{f,(i)})\}|y_t)p(x_t^f|x_{t-1}^{(i)})dx_t^f|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|, \tag{2.8}$$

which gives us the importance weight under $p(x_{0:t}|y_{1:t})$ without any augmented state space as follows:

$$w_t^{(i)} \quad = \quad \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})} w_{t-1}^{(i)}. \tag{2.9}$$

The integral in (2.8) might not be solved analytically. In order to estimate it, we can implement the naive Monte Carlo algorithm for each particle i as

$$\hat{q}(x_t^{(i)}|y_t, x_{t-1}^{(i)}) = \frac{1}{k}\sum\nolimits_{j=1}^{M} q_l(\hat{\Sigma}_t\{(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})x_t^{(i)} - \hat{Q}_t^{-1}x_t^{f,(j)})\}|y_t)|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|,$$

where $x_t^{f,(j)}$ is the sample from $p(x_t^f|x_{t-1}^{(i)})$ for $j = 1, \ldots, M$. So, the approximated weight would be

$$\hat{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{\hat{q}(x_t^{(i)}|y_t, x_{t-1}^{(i)})}\hat{w}_{t-1}^{(i)}. \tag{2.10}$$

The proposal density $q(x_t^{(i)}|y_t, x_{t-1}^{(i)})$ is approximated by the naive Monte Carlo method with sample size M. For N particles, we need to evaluate $q_l(\hat{\Sigma}_t\{(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})x_t^{(i)} - \hat{Q}_t^{-1}x_t^{f,(j)})\}|y_t)$ totally $N \times M$ times, which is very computationally expensive. However, it can be avoided by considering the augmented state space and the joint proposal density in the importance weight computation.

We presented a certain type of linear combination to combine x_t^l and x_t^f. However, in the APF we have the freedom to choose any type of weighted sum. For example, if we believe the state equation is not informative, then we can inflate the variance components in $\hat{Q}_t$, so the final particle would be generated with more weight on $x_t^{l,(i)}$ which is the particle from the IPF proposal. This is one extreme case. Another extreme case is to inflate the variance components in $\hat{\Sigma}_t$ when the observation equation is not informative. In such case, our final particle would be close to $x_t^{f,(i)}$, which is the particle from the NPF proposal. Thus, NPFs and IPFs can be viewed as the two extreme cases of APFs which strike a balance between the information from two different sources.

2.2 Justification

Evaluating the proposal $q(x_t^{(i)}|y_t, x_{t-1}^{(i)})$ for each i is a time consuming job. However, it can be avoided by considering the proposal as a joint density

$$q(x_t^{(i)}, x_t^{f,(i)}|y_t, x_{t-1}^{(i)}) = q(x_t^{(i)}|y_t, x_t^{f,(i)})p(x_t^{f,(i)}|x_{t-1}^{(i)}).$$

The density $q(x_t^{(i)}|y_t, x_t^{f,(i)})$ can be described in two parts: 1) sampling $x_t^{l,(i)}$ from $q_l(x_t^l|y_t)$ and 2) combining $x_t^{l,(i)}$ with $x_t^{f,(i)}$ via Equation (2.2). Notice that in our sampling procedure $x_t^{l,(i)}$ depends only on y_t, and $x_t^{f,(i)}$ would be treated as a constant in $q(\cdot|y_t, x_t^{f,(i)})$, so the conditional distribution of $x_t^{(i)}|y_t, x_t^{f,(i)}$ can be reduced to the conditional distribution of $x_t^{l,(i)}|y_t$. The proposal $q(x_t^{(i)}|y_t, x_t^{f,(i)})$ can be written as

$$\begin{aligned} q(x_t^{(i)}|y_t, x_t^{f,(i)}) &= q\{(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}(\hat{\Sigma}_t^{-1}x_t^{l,(i)} + \hat{Q}_t^{-1}x_t^{f,(i)})|y_t, x_t^{f,(i)}\} \qquad (2.11)\\ &= q_l((\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}x_t^{l,(i)}|y_t)\\ &= q_l(x_t^{l,(i)}|y_t)|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|\\ &\propto q_l(x_t^{l,(i)}|y_t). \end{aligned}$$

Therefore, $q(x_t^{(i)}|y_t, x_t^{f,(i)})$ can be evaluated to be proportional to $q(x_t^{l,(i)}|y_t)$, and the normalizing constant $|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|$ is the same for all particles $x_t^{(i)}$'s. The evaluation of $\hat{H}_t$, $\hat{R}_t$, and $\hat{Q}_t$ affect the combination of the particles from the two equations, but their values are not involved in the weight computation.

Now we illustrate how the augmented state space, $x_{1:t}^f$ would change the target density. Recall given x_{t-1}, x_t^f is free of any other state vectors or observations, and its relation with x_{t-1} is determined through the augmented state equation $p(x_t^f|x_{t-1})$. Thus, we have

$$\begin{aligned} p(x_{0:t}, x_{1:t}^f|y_{1:t}) &= \frac{p(x_{0:t}, x_{1:t}^f, y_{1:t})}{p(y_t|y_{1:t-1})p(y_{1:t-1})}\\ &= \frac{p(y_t|x_t)p(x_t|x_{t-1})p(x_t^f|x_{t-1})}{p(y_t|y_{1:t-1})}\frac{p(x_{0:t-1}, x_{1:t-1}^f, y_{1:t-1})}{p(y_{1:t-1})}\\ &= \frac{p(y_t|x_t)p(x_t|x_{t-1})p(x_t^f|x_{t-1})}{p(y_t|y_{1:t-1})}p(x_{0:t-1}, x_{1:t-1}^f|y_{1:t-1}). \end{aligned}$$

By augmenting $x_{1:t}^f$, we have additional component $p(x_t^f|x_{t-1})$ in the target density, but this part would cancel out in the importance weight evaluation as shown in (2.9).

$$\begin{aligned} w_t^{(i)} &= \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})p(x_t^{f,(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})}w_{t-1}^{(i)}\\ &= \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})p(x_t^{f,(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_t^{f,(i)})p(x_t^{f,(i)}|x_{t-1}^{(i)})}w_{t-1}^{(i)}\\ &= \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q_l(x_t^{l,(i)}|y_t)|(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}\hat{\Sigma}_t^{-1}|}w_{t-1}^{(i)}\\ &\propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q_l(x_t^{l,(i)}|y_t)}w_{t-1}^{(i)}. \end{aligned}$$

Recall that we construct our particle $x_t^{(i)}$ by taking a linear combination through estimates $\hat{H}_t$,$\hat{R}_t$, and $\hat{Q}_t$ from our samples, but those estimates only implicitly appear in the importance weight computation through $x_t^{(i)}$. The benefit from the above evaluation is that we avoid the potential increase in the computational cost as appeared in (2.9) and (2.10), and the importance weight is still exact up to a normalizing constant without any approximation. Therefore, we obtain the weighted sample $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ for the posterior distribution $p(x_{0:t}|y_{1:t})$.

As an optional step in the APF, we may perform resampling $x_t^{l,(i)}$ with probability proportional to $w_t^{l,(i)}$. In such case, $q(x_t^{(i)}|y_t, x_t^{f,(i)})$ would be approximately proportional to the observation density $p(y_t|x_t^{l,(i)})$ instead of $q_l(x_t^{l,(i)}|y_t)$, and we need to substitute $q_l(x_t^{l,(i)}|y_t)$ with $p(y_t|x_t^{l,(i)})$ in (2.9).

2.3 State Space Models with Additive Errors

In this section, we explain the APF algorithm for the state space model with additive error terms. We do not restrict our model to have normal errors. The following equations illustrate our model:

$$\begin{cases} y_t|x_t = h_t(x_t) + u_t, \ u_t \sim (0, R_t) \\ x_t|x_{t-1} = f_t(x_{t-1}) + v_t, \ v_t \sim (0, Q_t), \end{cases} \tag{2.12}$$

where (μ, Σ) denote a distribution with mean μ and variance Σ. For the APF with the general SSM in Section 2.1, evaluating variances in the SSM by $\hat{R}_t$ and $\hat{Q}_t$ is necessary to measure the amount of information in the observation and the state space evolution over time. Here in Model (2.12), we simply set $\hat{R}_t = R_t$ and $\hat{Q}_t = Q_t^f$, where Q_t^f is a variance of the augmented state noise, because we have $Var(h_t(\tilde{x}_t)+u_t|\tilde{x}_t) = Var(u_t)$ and $Var(f_t(\bar{x}_{t-1}) + v_t^f|\bar{x}_{t-1}) = Var(v_t^f)$.

The detailed algorithm is given as follows. At the initial step $t = 0$, draw $x_0^{(i)}$ from $q(x_0)$ for $i = 1, \ldots, N$, and compute the importance weight as

$$w_0^{(i)} = \frac{p(x_0^{(i)})}{q(x_0^{(i)})}.$$

For $i = 1, \ldots, T$, we repeat the following steps:

1. Draw $x_t^{f,(i)}$ from $p(x_t^f|x_{t-1}^{(i)})$.

2. Draw $x_t^{l,(i)}$ from a proposal density $q_l(x_t^l|y_t)$.

3. Let $\hat{H}_t$ denote the derivative of $h_t(x_t)$ with respect to x_t at $\tilde{x}_t$.

4. Let $\hat{\Sigma}_t = (\hat{H}_t' R_t^{-1} \hat{H}_t)^{-1}$. Then, combine the two particles from Steps 1 and 2 as

$$\begin{aligned} x_t^{(i)} &= (\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}(\hat{\Sigma}_t^{-1} x_t^{l,(i)} + \hat{Q}_t^{-1} x_t^{f,(i)}) \\ &= \hat{Q}_t(\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{l,(i)} + \hat{\Sigma}_t(\hat{\Sigma}_t + \hat{Q}_t)^{-1} x_t^{f,(i)}. \end{aligned} \tag{2.13}$$

5. Calculate the weight as
$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q_l(x_t^{l,(i)}|y_t)} w_{t-1}^{(i)}.$$

For the SSM with additive errors, $\hat{R}_t$ and $\hat{Q}_t$ are given in the model, and the estimation of $\hat{H}_t$ requires less computational cost than that for the general SSM in Section 2.1. In high–dimensional SSMs, the Monte Carlo estimation step of the coefficients in (2.3) could be unstable. The APF for SSMs with the additive errors is capable of avoiding that step.

2.3.1 SSMs with Gaussian Additive Noise and Linear Observation Equations

In this section, we consider a subclass of additive SSMs in Section 2.3. The model in (2.14) has Gaussian additive errors, and the observation equation has a linear operator H_t. In this type of SSMs, the OPF can be implemented, and the APF proposal distribution can be marginalized to the OPF proposal distribution when the model is not under–determined, that is, the dimension of y_t is less than or equal to the dimension of x_t.

$$\begin{cases} y_t|x_t = H_t x_t + u_t, \; u_t \sim N(0, R_t) \\ x_t|x_{t-1} = f_t(x_{t-1}) + v_t, \; v_t \sim N(0, Q_t). \end{cases} \tag{2.14}$$

Notice that in the linear Gaussian SSM, we have H_t as the derivative of $h_t(x_t)$. From Section 2.3, we have $\hat{R}_t = R_t$ and $\hat{Q}_t = Q_t^f$.

Here we present the APF algorithm. At the initial step $t = 0$, draw $x_0^{(i)}$ from $q(x_0)$ for $i = 1, \ldots, N$, and compute the importance weight as
$$w_0^{(i)} = \frac{p(x_0^{(i)})}{q(x_0^{(i)})}.$$

For $t = 1, \ldots, T$, we repeat the following steps.

1. Draw $x_t^{f,(i)}$ by $x_t^{f,(i)} = f_t(x_{t-1}^{(i)}) + v_t^{(i)}$ where $v_t^{(i)} \sim N(0, \hat{Q}_t)$.

2. Draw $x_t^{l,(i)}$ from a proposal distribution $q_l(x_t^l|y_t)$. Let $\Sigma_t = (H_t' R_t^{-1} H_t)^{-1}$. Here the proposal distri-

bution, which is proportional to $p(y_t|x_t)$, would be

$$N(H_t'(H_tH_t')^{-1}y_t, \Sigma_t). \tag{2.15}$$

3. Construct the particles by combining particles from the last two steps:

$$x_t^{(i)} = (\Sigma_t^{-1} + \hat{Q}_t^{-1})^{-1}(\Sigma_t^{-1}x_t^{l,(i)} + \hat{Q}_t^{-1}x_t^{f,(i)}). \tag{2.16}$$

4. The importance weights can be obtained by

$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q_l(x_t^{l,(i)}|y_t)}w_{t-1}^{(i)}.$$

Here are some remarks on the above algorithm.

1. In Step 2, $(H_t'R_t^{-1}H_t)$ might not be invertible if the dimension of x_t is less than the dimension of y_t. Thus, in the under–determined SSMs, we cannot choose $q_l(x_t^l|y_t)$ to be proportional to $p(y_t|x_t)$. However, we do not take this case into account because we assume it is always possible to have over–determined SSMs by adding artificial noised observations into the observation equation. For example, spatial smoothness conditions for x_t can be incorporated into the observation equation with small noise.

2. The OPF can be implemented for both over and under–determined SSMs without any modification. At time t, its proposal distribution would be

$$q(x_t^{(i)}|y_t, x_{t-1}^{(i)}) = N(\mu_t, \Sigma_t^*), \tag{2.17}$$

where $\Sigma_t^* = (\Sigma_t^{-1} + Q_t^{-1})^{-1}$, and $\mu_t = \Sigma_t^*(H_t'R_t^{-1}y_t + Q_t^{-1}f_t(x_{t-1}))$. Also, its importance weight can be obtained recursively as follows.

$$\begin{aligned} w_t^{(i)} &= p(y_t|x_{t-1}^{(i)})w_{t-1}^{(i)} \\ &\propto \exp\{\frac{1}{2}(y_t - H_tf_t(x_{t-1}^{(i)}))(R_t + H_tQ_tH_t')^{-1}(y_t - H_tf_t(x_{t-1}^{(i)}))'\}w_{t-1}^{(i)}. \end{aligned} \tag{2.18}$$

3. The importance weight of $x_t^{(i)}$ for the marginalized proposal density $q(x_t^{(i)}|y_t, x_{t-1}^{(i)})$ can be analytically

evaluated without the augmented state space as

$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})} w_{t-1}^{(i)}.$$

The marginalized proposal density would be

$$q(x_t^{(i)}|y_t, x_{t-1}^{(i)}) = N(\mu_t, \Sigma_t^*). \tag{2.19}$$

The last remark indicates that the two sets of particles from the APF and OPF are equivalent in distribution. We deliver this result as the following proposition.

Proposition 2.3.1. *For the SSM in (2.14) with dim(y_t) $\geq$ dim(x_t), if the APF utilizes the likelihood proposal in (2.15) and sets $v_t^f \overset{d}{=} v_t$, then the marginal proposal distribution of $x_t^{(i)}$ in the APF proposal is the same as the OPF proposal distribution (2.19).*

Proof. It is sufficient to prove that at any time t, the marginalized proposal distribution of the APF is $N(\mu_t, \Sigma_t^*)$ which is the OPF proposal distribution. Note that the conditional distribution of $x_t^{f,(i)}$ given $x_{t-1}^{(i)}$ follows $N(f_t(x_{t-1}^{(i)}), Q_t)$, and the conditional distribution of $x_t^{l,(i)}$ given y_t follows $N(H_{t'}(H_t H_{t'})^{-1} y_t, \Sigma_t)$. The derivation of the conditional moments of $x_t^{(i)}$ given y_t and $x_{t-1}^{(i)}$ is as follows:

$$\begin{aligned}
E(x_t^{(i)}|y_t, x_{t-1}^{(i)}) &= E((\Sigma_t^{-1} + Q_t^{-1})^{-1}(\Sigma_t^{-1} x_t^{l,(i)} + Q_t^{-1} x_t^{f,(i)})|y_t, x_{t-1}^{(i)}) \\
&= (\Sigma_t^{-1} + Q_t^{-1})^{-1}\{H_t{}' R_t^{-1} H_t H_{t'}(H_t H_{t'})^{-1} y_t + Q_t^{-1} f_t(x_{t-1}^{(i)})\} \\
&= (\Sigma_t^{-1} + Q_t^{-1})^{-1}\{H_t{}' R_t^{-1} y_t + Q_t^{-1} f_t(x_{t-1}^{(i)})\} \\
&= \mu_t
\end{aligned}$$

$$\begin{aligned}
Var(x_t^{(i)}|y_t, x_{t-1}^{(i)}) &= Var((\Sigma_t^{-1} + Q_t^{-1})^{-1}(\Sigma_t^{-1} x_t^{l,(i)} + Q_t^{-1} x_t^{f,(i)})|y_t, x_{t-1}^{(i)}) \\
&= (\Sigma_t^{-1} + Q_t^{-1})^{-1}\{\Sigma_t^{-1}\Sigma_t\Sigma_t^{-1} + Q_t^{-1} Q_t Q_t^{-1}\}(\Sigma_t^{-1} + Q_t^{-1})^{-1} \\
&= (\Sigma_t^{-1} + Q_t^{-1})^{-1} \\
&= \Sigma_t^*
\end{aligned}$$

Thus, μ_t and Σ_t^* are conditional mean and variance given y_t and x_{t-1} of our final particle $x_t^{(i)}$. Since the conditional distribution of $x_t^{l,(i)}$ and $x_t^{f,(i)}$ given y_t and x_{t-1} follow independent normal distributions, the weighted sum of the two random vectors follow a normal distribution. Therefore, the marginalized proposal density for APFs is the same as the proposal density for OPFs. □

Next, it might be interesting to see how the APF and the OPF are related in terms of their importance weights. If the APF is implemented for the over–determined SSM in (2.14), then the variance of its importance weight cannot be smaller than the variance of the OPF weight. See Doucet et al. (2001) for the detail.

2.4 Simulation Studies

The simulations were coded in MATLAB and ran on a UNIX machine with a 2.40GHz processor.

2.4.1 Nonlinear Filtering

When the observation equation is nonlinear, the optimal particle filter is not available in general. The naive particle filter and the independent particle filter are often used to estimate $E(X_t|Y_{1:t})$. We want to compare the performance of three methods: the augmented particle filter, the naive particle filter, and the independent particle filter on the following nonlinear SSM in Gordon et al. (1993):

$$\begin{cases} y_t|x_t = x_t^2/20 + u_t \\ x_t|x_{t-1} = 0.5x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2(t-1)) + v_t, \end{cases} \tag{2.20}$$

where $v_t \sim N(0, \sigma^2)$ and $u_t \sim N(0, \delta^2)$. Because of the difficulty in deriving the proposal density of x_t^l based on the observation density

$$p(y_t|x_t) \propto \exp\left(-\frac{1}{2\delta^2}(\frac{x_t^2}{20} - y_t)^2\right),$$

we use the proposal distribution $q(x_t|y_t)$ suggested by Lin et al. (2005), which can be obtained by plugging in the linearization of $h(x_t)$ into $p(y_t|x_t)$,

$$q_l(x_t^l|y_t) = \begin{cases} 0.5N(c, s^2) + 0.5N(-c, s^2) & y_t > 0 \\ N(0, 25\delta^2) & y_t \le 0, \end{cases} \tag{2.21}$$

where $c = \sqrt{20y_t}$, and $s^2 = \min(5\delta^2/y_t, 25\delta^2)$. The model has additive error terms, so the APF in Section 2.3 can be implemented. The three coefficients for the linear combination are obtained as $\hat{H}_t = \frac{1}{10}\tilde{x}_t$, $\hat{R}_t = \delta^2$, and $\hat{Q}_t = \sigma^2$, where $\tilde{x}_t = c$ if $y_t > 0$; 0 otherwise. For each t, the particle from the APF is

$$x_t^{(i)} = (\frac{\tilde{x}_t^2}{100\delta^2} + \frac{1}{\sigma^2})^{-1}(\frac{\tilde{x}_t^2}{100\delta^2}x_t^{l,(i)} + \frac{1}{\sigma^2}x_t^{f,(i)}),$$

where $x_t^{f,(i)}$ is the sample from $p(x_t|x_{t-1}{}^{(i)})$. Thus, if $y_t > 0$, the APF proposal would be simplified to the NPF proposal. The IPF proposal density is the same as the density in (2.21), and no multiple matching is considered. The proposal density for the NPF is given as the state equation in (2.20). The importance weights are computed recursively for the three methods.

$$
\begin{aligned}
APF: w_t^{(i)} &= \begin{cases} \dfrac{N(y_t|\frac{(x_t^{(i)})^2}{20},\delta^2)N(x_t^{(i)}|0.5x_{t-1}^{(i)}+\frac{25x_{t-1}^{(i)}}{1+(x_{t-1}^{(i)})^2}+8\cos(1.2(t-1)),\sigma^2)}{q_l(x_t^{l,(i)}|y_t)} w_{t-1}^{(i)} & y_t > 0 \\ N(y_t|\frac{(x_t^{(i)})^2}{20},\delta^2)w_{t-1}^{(i)} & y_t \le 0 \end{cases} \\
IPF: w_t^{(i)} &= \frac{N(y_t|\frac{(x_t^{l,(i)})^2}{20},\delta^2)N(x_t^{l,(i)}|0.5x_{t-1}^{(i)}+\frac{25x_{t-1}^{(i)}}{1+(x_{t-1}^{(i)})^2}+8\cos(1.2(t-1)),\sigma^2)}{q_l(x_t^{l,(i)}|y_t)} w_{t-1}^{(i)} \\
NPF: w_t^{(i)} &= N(y_t|\frac{(x_t^{(i)})^2}{20},\delta^2)w_{t-1}^{(i)},
\end{aligned}
$$

where $N(x|\mu,\sigma^2)$ denote the normal density w.r.t. x with mean μ and variance σ^2.

We consider sixteen different combinations of $\delta \in \{1/4, 1/2, 1, 2\}$ and $\sigma \in \{1, 2, 4, 8\}$. The number of particles is set to be 100. Each method is implemented with two schemes: 1) resample at every time step with the time step $T = 100$ and 2) no resampling with $T = 5$. We chose a very small T for the second scheme because otherwise, the naive particle filter would crash too often to obtain stable RMSEs for some settings.

To measure the performance of each method, we compute the root mean square error (RMSE) and the standard error of the RMSE (se(RMSE)). Let $\hat{X}_{0:T}$ denote $(\hat{E}(X_0|Y_1), \hat{E}(X_1|Y_{1:2}), \ldots, \hat{E}(X_T|Y_{1:T}))$. We have

$$
\text{RMSE} = \frac{1}{K}\sum_{k=1}^{K} \sqrt{\frac{1}{T}||\hat{X}_{0:T}^k - X_{0:T}^k||^2},
$$

$$
\text{se(RMSE)} = \sqrt{\frac{1}{K}\widehat{Var}\left(\sqrt{\frac{1}{T}||\hat{X}_{0:T}^k - X_{0:T}^k||^2}\right)},
$$

where $K = 10,000$ is the number of independent repeated experiments.

The results for no resampling case are presented in Table 2.2. We can see that for small observation noise δ and large state noise σ, the APF outperforms the NPF. For large δ and small σ, the APF is slightly worse than the NPF, but the difference is small. The reason is that the APF uses the information from both the observation and state equations to construct the proposal distribution, but the NPF only uses the state equation. Notice that using the observation equation alone for constructing proposal densities is not very effective, as shown in the results for the IPF. Also, we present the results without resampling in Table 2.1. From the table, we can see the similar pattern as that in Table 2.2, but in many settings NPFs crash due to

Table 2.1: No resampling: The RMSEs and their standard errors of the augmented particle filter (APF), the naive particle filter (NPF), and the independent particle filter (IPF) for different combinations of δ and σ with $T = 5$.

		$\sigma = 1$			$\sigma = 2$	
	APF	NPF	IPF	APF	NPF	IPF
$\delta = 1/4$	2.5323	N/A	7.6962	2.9654	N/A	7.4247
	(0.0281)		(0.0223)	(0.0239)		(0.0258)
$\delta = 1/2$	2.8375	2.4707	7.6724	3.1192	3.2822	7.3664
	(0.0307)	(0.0239)	(0.0213)	(0.0251)	(0.0263)	(0.0250)
$\delta = 1$	3.3964	2.6153	7.4040	3.6195	3.3728	7.1542
	(0.0326)	(0.0217)	(0.0232)	(0.0284)	(0.0238)	(0.0247)
$\delta = 2$	4.3756	2.9464	7.2872	4.3674	3.5170	7.1231
	(0.0356)	(0.0217)	(0.0239)	(0.0311)	(0.0218)	(0.0241)
		$\sigma = 4$			$\sigma = 8$	
	APF	NPF	IPF	APF	NPF	IPF
$\delta = 1/4$	4.6565	N/A	7.5133	8.8570	N/A	10.4759
	(0.0271)		(0.0310)	(0.0476)		(0.0529)
$\delta = 1/2$	4.7176	N/A	7.5175	8.9116	N/A	10.5194
	(0.0273)		(0.0310)	(0.0477)		(0.0535)
$\delta = 1$	4.9248	5.3864	7.4968	8.9539	N/A	10.5390
	(0.0284)	(0.0334)	(0.0305)	(0.0478)		(0.0533)
$\delta = 2$	5.4727	5.3699	7.6135	9.1470	10.5610	10.5754
	(0.0307)	(0.0301)	(0.0296)	(0.0486)	(0.0675)	(0.0523)

the degeneracy of the importance weights.

2.4.2 Maneuvering Target Tacking

We test the APF on a multi–dimensional nonlinear SSM. The tracking problem given in Ikoma et al. (2001) and Lin et al. (2005) is aimed to track a maneuvering target (e.g. ship or aircraft) over time ϕ in seconds. The dynamics of the target is given as a differential equation, and the discretization of the model is as follows. (see Ikoma et al. (2001) for the detail.)

$$\xi_{\phi+\Delta\phi} = A(\Delta\phi)\xi_\phi + B(\Delta\phi)v_\phi, \tag{2.22}$$

Table 2.2: Resampling at every time step: The RMSEs and their standard errors of the augmented particle filter (APF), the naive particle filter (NPF), and the independent particle filter (IPF) for different combinations of δ and σ with $T = 100$.

		$\sigma = 1$			$\sigma = 2$	
	APF	NPF	IPF	APF	NPF	IPF
$\delta = 1/4$	3.8377	3.535	7.0362	3.9126	4.525	7.5077
	(0.0170)	(0.012)	(0.0056)	(0.0116)	(0.013)	(0.0063)
$\delta = 1/2$	4.2076	3.334	7.1632	4.0988	4.192	7.4909
	(0.0174)	(0.010)	(0.0051)	(0.0116)	(0.011)	(0.0060)
$\delta = 1$	4.7366	3.394	7.3042	4.4781	4.151	7.4682
	(0.0173)	(0.008)	(0.0044)	(0.0122)	(0.008)	(0.0054)
$\delta = 2$	5.4820	3.693	7.5929	5.1177	4.423	7.6222
	(0.0164)	(0.007)	(0.0036)	(0.0124)	(0.006)	(0.0044)
		$\sigma = 4$			$\sigma = 8$	
	APF	NPF	IPF	APF	NPF	IPF
$\delta = 1/4$	5.0669	6.967	8.1862	8.9337	12.81	11.1696
	(0.0088)	(0.019)	(0.0082)	(0.0139)	(0.027)	(0.0138)
$\delta = 1/2$	5.1639	6.155	8.1684	8.9693	11.729	11.1754
	(0.0087)	(0.016)	(0.0080)	(0.0143)	(0.026)	(0.0139)
$\delta = 1$	5.3603	5.775	8.1336	9.0266	10.752	11.1822
	(0.0089)	(0.013)	(0.0079)	(0.0142)	(0.024)	(0.0140)
$\delta = 2$	5.7650	5.778	8.1492	9.1098	10.091	11.1450
	(0.0092)	(0.009)	(0.0072)	(0.0142)	(0.021)	(0.0138)

where

$$A(\Delta\phi) = \begin{bmatrix} 1 & 0 & \Delta\phi & 0 & a_1 & 0 \\ 0 & 1 & 0 & \Delta\phi & 0 & a_1 \\ 0 & 0 & 1 & 0 & a_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & a_2 \\ 0 & 0 & 0 & 0 & e^{-\alpha\Delta\phi} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha\Delta\phi} \end{bmatrix}, \tag{2.23}$$

$$B(\Delta\phi) = \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & b_1 & 0 & b_2 & 0 & b_3 \end{bmatrix}'.$$

The components in (2.23) are as follows:

$$b_1 = \frac{1}{\alpha}\left(\frac{(\Delta\phi)^2}{2} - a_1\right),$$

$$a_1 = b_2 = \frac{1}{\alpha}(\Delta\phi - a_2),$$

$$a_2 = b_3 = \frac{1}{\alpha}(1 - e^{-\alpha\Delta\phi}).$$

Here ξ_ϕ has six components: the first two components $\xi_{\phi,1:2}$ for the position, the middle two $\xi_{\phi,3:4}$ for the velocity, and the last two $\xi_{\phi,5:6}$ for the acceleration of the target at time ϕ in the Cartesian space. As in Ikoma et al. (2001), we assume an independent Cauchy state noise $v_\phi = (v_{\phi,1}, v_{\phi,2})'$ with the density

$$p(v_{\phi,i}) = \frac{q}{\pi(v_{\phi,i}^2 + q^2)}. \tag{2.24}$$

For notational convenience, let x_t denote $\xi_{t\Delta\phi}$. In terms of x_t, state dynamics can be rewritten as

$$x_t = A(\Delta\phi)x_{t-1} + B(\Delta\phi)v_t, \tag{2.25}$$

where v_t has the same density in (2.24). The target position is measured by a radar, so the observation at time t is the measurement of angle and distance of the target from the origin.

$$y_t = h(x_t) + u_t, \tag{2.26}$$

where

$$h(x_t) = \left(\arctan\left(\frac{x_{t,1}}{x_{t,2}}\right), \sqrt{x_{t,1}^2 + x_{t,2}^2}\right)'. \tag{2.27}$$

As in Ikoma et al. (2001), we assume the Gaussian observation noise

$$u_t \sim N(0, R),\ \ R = \sigma \begin{bmatrix} 10^{-10} & 0 \\ 0 & 10^{-2} \end{bmatrix}. \tag{2.28}$$

The initial distribution at $\phi = 0$ of the position, velocity, and acceleration of the target is

$$N((50000, 5000, 0, 10, 0, 0)', \mathbf{I}_6).$$

We sample the true trajectory of the target from (2.22) with $\Delta\phi = 0.01$. The observations arrive every 3.75 seconds, so for the implementation we use the dynamics in (2.25) with $\Delta\phi = 3.75$. We track the target for the first 375 seconds, so the total time step T is 100. We set $\alpha = 1000$ in (2.23), $q = 1$ in (2.24), and the number of particles $N = 1000$. We consider three different values of $\sigma \in \{0.01, 1, 100\}$. In each setting, the experiment is repeated 100 times. We compare three methods with resampling at every step: the augmented particle filter, the independent particle filter, and the naive particle filter. An extension of the IPF (MIPF-1, see Lin et al. (2005)) is considered with no multiple matching, and its proposal distribution for the position vector is the same as in (2.29).

Given that the state noise is Cauchy, we truncate the density in (2.24) on its 99.99% highest density region to estimate $\hat{Q}$. The naive Monte Carlo method with sample size 10,000 is used to obtain $\hat{Q}$. The proposal distribution for the observation equation is

$$q_l(x^l_{t,1:2}|y_t) = N((\tilde{x}_{t,1}, \tilde{x}_{t,2})', \hat{\Sigma}_t), \tag{2.29}$$

where

$$\begin{aligned} &(\tilde{x}_{t,1}, \tilde{x}_{t,2})' = h^{-1}(y_t), \\ &\hat{\Sigma}_t^{-1} = \mathbf{I}_2 \circ (\hat{H}_t' R^{-1} \hat{H}_t). \end{aligned}$$

Here $\circ$ denotes the component–wise matrix product. The matrix $\hat{\Sigma}_t^{-1}$ is tapered to prevent a potential matrix inversion problem. The entities in $\hat{H}_t$, which is the derivative of $h(\cdot)$ at the mode of $p(y_t|x_t)$, is as follows:

$$\hat{H}_t = \begin{bmatrix} \frac{1}{1+\left(\frac{\tilde{x}_{t,1}}{\tilde{x}_{t,2}}\right)^2} \frac{1}{\tilde{x}_{t,2}} & -\frac{1}{1+\left(\frac{\tilde{x}_{t,1}}{\tilde{x}_{t,2}}\right)^2} \frac{\tilde{x}_{t,1}}{\tilde{x}^2_{t,2}} \\ \left(\tilde{x}^2_{t,1} + \tilde{x}^2_{t,2}\right)^{-0.5} \tilde{x}_{t,1} & \left(\tilde{x}^2_{t,1} + \tilde{x}^2_{t,2}\right)^{-0.5} \tilde{x}_{t,2} \end{bmatrix}. \tag{2.30}$$

Since the distribution of the state noise is heavy–tailed, we relaxed the density of the augmented state vector to cover the tail region with higher probabilities. Also, the augmentation is only for the position vector $x_{t,1:2}$. The augmented state density $p(x^f_t|x_{t-1})$ follows the dynamics in (2.22), but we set $q = 1000$ for the density of Cauchy noise in (2.24). The observation equation relies only on the position vector, so the sampling needs two steps:

1. We sample the position vector $x^{(i)}_{t,1:2}$ by combining $x^{l,(i)}_{t,1:2}$ from the density in (2.29) and $x^{f,(i)}_{t,1:2}$ from $p(x^f_t|x^{(i)}_{t-1})$ through the linear combination in (2.2).
2. Given $x^{(i)}_{t,1:2}$ and $x^{(i)}_{t-1}$, the velocity and acceleration $x^{(i)}_{t,3:6}$ follows a degenerated distribution. So, we directly sample from $p(x_{t,3:6}|x^{(i)}_{t,1:2}, x^{(i)}_{t-1})$.

The APF importance weight at t is

$$w^{(i)}_t = \frac{N(y_t|h(x^{(i)}_{t,1:2}), R) p(x^{(i)}_{t,1:2}|x^{(i)}_{t-1})}{N(x^{l,(i)}_{t,1:2}|h^{-1}(y_t), \hat{\Sigma}_t)} w^{(i)}_{t-1} \tag{2.31}$$

The results are given in Table 2.3. The APF has smaller RMSEs than the other two methods. Balancing the information from the past particles and the current observation in the APF works well for the tracking problem.

Table 2.3: The RMSEs and their standard errors of three filtering methods for the model in (2.22): the augmented particle filter (APF), the independent particle filter (IPF), and the naive particle filter (NPF).

$\sigma = 0.01$	RMSE	se(RMSE)	CPU Time (sec)
APF:	1.161	0.337	2.919
IPF:	2.840	0.338	2.257
NPF:	4784.008	1193.049	2.029
$\sigma = 1$	RMSE	se(RMSE)	CPU Time (sec)
APF:	0.801	0.079	3.015
IPF:	4.526	0.229	2.414
NPF:	1139.173	366.924	2.117
$\sigma = 100$	RMSE	se(RMSE)	CPU Time (sec)
APF:	7.934	0.172	2.942
IPF:	42.328	2.149	2.286
NPF:	67.595	39.252	2.029

2.4.3 Linear Gaussian Models

For the simulation study, we consider a slightly high dimensional SSM as follows:

$$\begin{cases} y_t|x_t = Hx_t + u_t, u_t \sim N(0, R), \\ x_t|x_{t-1} = x_{t-1} + v_t, v_t \sim N(0, Q). \end{cases} \tag{2.32}$$

The dimension of x_t and the dimension of y_t are set to be 50, and the time step is considered up to $T = 300$. The matrix H is very sparse, which is often the case in the high dimensional SSM. In many SSMs, how the state vector x_t evolves over time is often unknown, so the state equation is often chosen to be a random walk. The matrix H is randomly chosen as follows: for each row, we randomly select centers without replacement. If the center of a certain row is in the middle column, then the corresponding row of H will have three nonzero consecutive components at the center. If the center is at the edge of the matrix, then we will assign only two nonzero consecutive components around the center. The values of each nonzero component is drawn from the standard normal distributions. The matrix H used for the experiments is given in Figure 2.2.

The covariance matrix Q is assumed to be a banded matrix with 4 on diagonal, 0.5 on the first off–diagonal, 0.1 on the second off–diagonal, and all other components are 0. We assume the covariance matrix R is $\frac{1}{4} \cdot \mathbf{I}_{50}$ where the subscript 50 indicates the dimension of the given identity matrix. The number of particles N for each experiment is set to be 100, and the experiment is repeated for $K = 1,000$ times.

We compared five methods: 1) the augmented particle filter (APF) discussed in Section 2.3.1, 2) the optimal particle filter (OPF), 3) the ensemble Kalman filter (EnKF), 4) the independent particle filter (IPF), and 5) the naive particle filter (NPF). Under the current model, the IPF and NPF crash too often to obtain stable results, so we present their results only with resampling. Note that the target density of the

Table 2.4: No Resampling: The RMSEs and their standard errors of three filtering methods for the model in (2.32): the augmented particle filter (APF), the optimal particle filter (OPF), and the ensemble Kalman filter (EnKF). For the APF and the OPF, we provide CV^2 to compare their variances of importance weights. $CV^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}(w_T^{(i)} - \bar{w}_T)^2}{(\bar{w}_T)^2}$ where $\bar{w}_T = \frac{1}{N}\sum_{i=1}^{N} w_T^{(i)}$.

	RMSE	se(RMSE)	CV^2	se(CV^2)	CPU Time (sec)	# of Crashes
APF	38.704	0.364	95.841	10.092	11.539	0
OPF	42.462	0.382	97.944	5.749	7.234	24
EnKF	46.097	0.533			3.577	0

Table 2.5: Resampling at every time step: The RMSEs and their standard errors of four filtering methods for the model in (2.32): the augmented particle filter (APF), the optimal particle filter (OPF), the independent particle filter (IPF), and the naive particle filter (NPF).

	RMSE	se(RMSE)	CPU Time
APF	35.870	0.410	11.942
OPF	41.602	0.458	7.537
IPF	941.912	8.519	5.488
NPF	101.322	0.505	2.683

APF is augmented with x_t^f.

In order to implement the APF under such a linear Gaussian model, we let $\hat{H}_t = H$, $\hat{R}_t = \frac{1}{4} \cdot \mathbf{I}_{50}$ and $\hat{Q}_t = Q$. The sampling procedure for the APF follows the algorithm described in Section 2.3.1 with $f_t(x_{t-1}) = x_{t-1}$. The proposal density based on the observation equation is chosen as

$$q_l(x_t^l|y_t) = N(H'(HH')^{-1}y_t, \frac{1}{4}(H'H)^{-1}),$$

which will also be used as the IPF proposal. We implemented the IPF with no multiple matching, so its importance weight is

$$w_t^{(i)} = N(x_t^{(i)}|x_{t-1}^{(i)}, Q)w_t^{(i)}.$$

The proposal density of the NPF is the state equation $N(x_t|x_{t-1}^{(i)}, Q)$, and its importance weight is $w_t^{(i)} = N(y_t|Hx_t^{(i)}, \frac{1}{4} \cdot \mathbf{I}_{50})w_{t-1}^{(i)}$. The proposal density of the OPF is

$$q(x_t^{(i)}|y_t, x_{t-1}^{(i)}) = N(x_t^{(i)}|\Sigma_t^*(\frac{1}{4} \cdot H_t'y_t + Q_t^{-1}x_{t-1}^{(i)}), \Sigma_t^*),$$

where $\Sigma_t^* = (4 \cdot H'H + Q_t^{-1})^{-1}$, and the importance weight would be evaluated as follows:

$$w_t^{(i)} = \exp\{\frac{1}{2}(y_t - Hx_{t-1}^{(i)})(\frac{1}{4} \cdot \mathbf{I}_{50} + HQH')^{-1}(y_t - Hx_{t-1}^{(i)})'\}w_{t-1}^{(i)}. \quad (2.33)$$

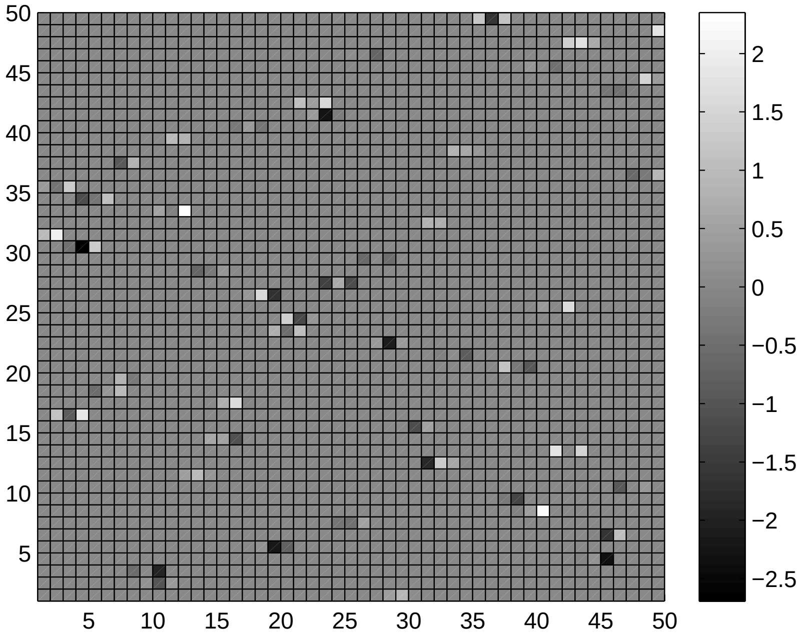

Figure 2.2: The sparse matrix H used for the data generation in Section 2.4.3. In the gray scale, the brighter component has a larger value than the darker components.

The results with and without resampling are given in Tables 2.4 and 2.5 respectively. We can see that APFs have significantly smaller RMSEs than the other methods. The IPF and NPF gave much larger RMSEs than the APF, so it could be very risky to utilize only one equation in SSMs to construct proposal densities. The simulation results also illustrate that with a relatively small number of particles, the EnKF without localization can be worse than the APF. The APF can be even better than the OPF for this linear Gaussian model. Also, the OPF without resampling crashes sometimes. Given that the particles from the APF and the OPF have the same marginal distributions, the augmented state space may improve the quality of the estimate.

2.4.4 Lorenz-96 Model

The Lorenz-96 model (Lorenz, 2006) provides a SSM continuous in time, but it has spatially discrete state space. The model is to study high–dimensional chaotic dynamics such as the atmosphere. The Lorenz-96 model is defined by a set of differential equations over time ϕ:

$$\frac{\xi_{\phi,i}}{d\phi} = (\xi_{\phi,(i+1 \bmod k)} - \xi_{\phi,(i-2 \bmod k)})\xi_{\phi,(i-1 \bmod k)} - \xi_{\phi,i} + F.$$

After linearizing Lorenz-96 model via the first order Euler method with time step $\Delta\phi = 0.001$, we have:

$$\xi_{\phi+\Delta\phi,i} = \Delta\phi(\xi_{\phi,i-1}(\xi_{\phi,i+1} - \xi_{\phi,i-2}) - \xi_{\phi,i} + F) + \xi_{\phi,i},\ i = 1, \dots, k \tag{2.34}$$

The spatial relationship in the state space is given as $\xi_{\phi,0} := \xi_{\phi,k}$, $\xi_{\phi,-1} := \xi_{\phi,k-1}$, and $\xi_{\phi,k+1} := \xi_{\phi,1}$. For notational convenience, let $x_{t+1,i}$ denote $\xi_{t\Delta\phi,i}$ which is the i–th spatial component of the discretized state vector at time $t \times \Delta\phi$. Then, we have the discretized nonlinear state equation after adding random perturbations to (2.34) as follows.

$$x_{t,i} = \Delta\phi(x_{t-1,i-1}(x_{t-1,i+1} - x_{t-1,i-2}) - x_{t-1,i} + F) + x_{t-1,i} + v_{t,i},\ i = 1, \dots, k \tag{2.35}$$

The dimension of x_t is $k = 80$, the constant $F = 8$, and $v_{t,i} \sim N(0, \Delta\phi)$. We consider two different observation equations:

1. full observation: $y_{t,i} = x_{t,i} + u_{t,i}$ for $i = 1, 2, ..., k$.

2. half observation: $y_{t,i} = x_{t,2i-1} + u_{t,i}$ for $i = 1, 2, ..., k/2$.

For the half observation case, we can examine the performance of the APF when the SSM is under–determined. Each $u_{t,i}$ is assumed to follow independent $N(0, 0.01)$ distribution. The data for each experiment are generated with the number of time steps $T = 500$. Six different SMC methods are implemented with the number of particles $N = 50$. For both the half and full observation cases, the OPF utilizes the proposal in (2.19), and its importance weight is

$$w_t^{(i)} = \exp\{\frac{1}{2}(y_t - Hx_{t-1}^{(i)})(R + HQH')^{-1}(y_t - Hx_{t-1}^{(i)})'\}w_{t-1}^{(i)}, \tag{2.36}$$

where for the half observation case $R + HQH' = (0.01 + 0.001) \cdot \mathbf{I}_{40}$, $Hx_{t-1} = x_{t-1,A}$ and for the full observation case $R+HQH' = (0.01+0.001)\cdot\mathbf{I}_{80}$, $Hx_{t-1} = x_{t-1}$. The NPF utilizes $p(x_t|x_{t-1})$ as the proposal distribution, and its weight is $w_t^{(i)} = N(y_t|x_t^{(i)}, 0.01\cdot\mathbf{I}_{dim(y_t)})w_{t-1}^{(i)}$. The LEnKF utilizes a covariance tapering matrix $C = \mathbf{I}_{80}$. See Butala et al. (2008) for the details about the LEnKF.

For the full observation case, the dimension of y_t is 80, and the proposal distribution for the likelihood is chosen to be $q_l(x_t^l|y_t) = N(y_t, 0.01 \cdot \mathbf{I}_{80})$ for the APF and the IPF. The importance weight of the IPF is $w_t^{(i)} = p(x_t^{(i)}|x_{t-1}^{(i)})w_{t-1}^{(i)}$. The APF combines particles as

$$x_t^{(i)} = (\frac{1}{0.01} + \frac{1}{0.001})^{-1}(\frac{1}{0.01}x_t^{l,(i)} + \frac{1}{0.001}x_t^{f,(i)}),$$

and the importance weight is

$$w_t^{(i)} = \frac{N(y_t|x_t^{(i)}, 0.01 \cdot \mathbf{I}_{80})p(x_t^{(i)}|x_{t-1}^{(i)})}{N(x_t^{l,(i)}|y_t, 0.01 \cdot \mathbf{I}_{80})} w_{t-1}^{(i)}$$

For the half observation case, the dimension of y_t is 40, and we can partition the state space into two parts: 1) $x_{t,A} := \{x_{t,2i-1}\}_{i=1,2,\dots,k/2}$ and 2) $x_{t,B} := \{x_{t,2i}\}_{i=1,2,\dots,k/2}$. The second state vector $x_{t,B}$ does not depend on y_t, and it is free of the first state vector $x_{t,A}$ given x_{t-1}. Hence, when we implement the APF, only $x_{t,A}$ will be updated by y_t, and $x_{t,B}$ will be updated through the state equation. That is, the final particle at time t would be $x_t^{(i)} = [x_{t,A}^{(i)}, x_{t,B}^{(i)}]$, where

$$\begin{aligned} x_{t,A}^{(i)} &= (\frac{1}{0.01} + \frac{1}{0.001})^{-1}(\frac{1}{0.01}x_{t,A}^{l,(i)} + \frac{1}{0.001}x_{t,A}^{f,(i)}), \\ x_{t,B}^{(i)} &= x_{t,B}^{f,(i)}. \end{aligned}$$

The likelihood proposal distribution for $x_{t,A}^l$ would be $q_l(x_{t,A}^l|y_t) = N(y_t, 0.01 \cdot \mathbf{I}_{40})$. Define a function

$$\delta_{x_{t,B}^f}(x_{t,B}) = \begin{cases} 1 & \text{if } x_{t,B} = x_{t,B}^f \\ 0 & \text{if } x_{t,B} \neq x_{t,B}^f \end{cases},$$

we have the proposal distribution for the APF as

$$\begin{aligned} & q(x_t, x_t^f|y_t, x_{t-1}) \\ &= q(x_t|y_t, x_t^f)p(x_t^f|x_{t-1}) \\ &= \delta_{x_{t,B}^f}(x_{t,B})q(x_{t,A}|y_t, x_{t,A}^f)p(x_t^f|x_{t-1}) \\ &\propto \delta_{x_{t,B}^f}(x_{t,B})q_l(x_{t,A}^l|y_t)p(x_t^f|x_{t-1}). \end{aligned}$$

Only the first part $x_{1:T,A}^f$ is augmented into the state space, so the target is $p(x_{1:T}, x_{1:T,A}^f|y_{1:T})$. Thus, the importance weight in each step is evaluated as follows:

$$\begin{aligned} w_t^{(i)} &= \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})p(x_{t,A}^{f,(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_t^{f,(i)})p(x_t^{f,(i)}|x_{t-1}^{(i)})} w_{t-1}^{(i)} \\ &= \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q_l(x_{t,A}^{l,(i)}|y_t)p(x_{t,B}^{f,(i)}|x_{t-1}^{(i)})} w_{t-1}^{(i)} \\ &= \frac{p(y_t|x_{t,A}^{(i)})p(x_{t,A}^{(i)}|x_{t-1}^{(i)})}{q_l(x_{t,A}^{l,(i)}|y_t)} w_{t-1}^{(i)}. \end{aligned}$$

Table 2.6: Resampling: The RMSEs and their standard errors of four filtering methods for the model in (2.35): the augmented particle filter(APF), the optimal particle filter (OPF), the independent particle filter (IPF), and the naive particle filter (NPF).

Half	RMSE	se(RMSE)	CPU Time (sec)
APF	7.202	0.033	38.741
OPF	7.316	0.035	46.506
IPF	7.641	0.037	22.365
NPF	6.905	0.030	15.638
Full	RMSE	se(RMSE)	CPU Time (sec)
APF	0.837	0.00020	92.306
OPF	0.737	0.00016	70.855
IPF	1.137	0.00008	43.195
NPF	1.300	0.00064	25.384

Table 2.7: No Resampling: The RMSE and their standard errors of four filtering methods for the model in (2.35). the augmented particle filter(APF), the optimal particle filter (OPF), the ensemble Kalman filter (EnKF), and the localized EnKF (LEnKF). $CV^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(w_T^{(i)}-\bar{w}_T\right)^2}{(\bar{w}_T)^2}$ where $\bar{w}_T = \frac{1}{N}\sum_{i=1}^{N} w_T^{(i)}$.

Half	RMSE	se(RMSE)	CV^2	se(CV^2)	CPU Time (sec)
APF	6.289	0.021	48.946	0.873	32.371
OPF	6.181	0.019	48.999	0.014	39.507
EnKF	7.589	0.037			16.834
LEnKF	5.297	0.025			16.027
Full	RMSE	se(RMSE)	CV^2	se(CV^2)	CPU Time (sec)
APF	0.898	0.00019	48.936	0.860	92.193
OPF	0.897	0.00017	48.720	2.178	69.633
EnKF	0.697	0.00019			30.091
LEnKF	0.473	0.00007			29.506

The proposal distribution for the IPF is chosen to be $N(x_{t,A}|y_t, 0.01\cdot\mathbf{I}_{40})p(x_{t,B}|x_{t-1})$, so the IPF importance weight is $w_t^{(i)} = p(x_{t,A}^{(i)}|x_{t-1}^{(i)})w_{t-1}^{(i)}$.

The RMSEs of six filtering methods based on $1,000$ independent experiments are presented in Table 2.6 (resampling at every time step) and Table 2.7 (no resampling). The NPF and IPF cannot be implemented without resampling, because they crash too many times due to the degeneracy of the importance weights. Note that the EnKF and LEnKF cannot guarantee the convergence of their estimates to $E(X_t|Y_{1:t})$ when the SSMs include non–linear functions as in (2.35). The APF and the OPF performed similarly.

Chapter 3

The Localized Augmented Particle Filter

Bengtsson et al. (2008) showed that the number of particles has to increase exponentially with the dimension of x_t for some SSMs in order to avoid the degeneracy of the importance weights in the PF. In practice, however, we cannot afford a large number of particles due to the computational constraints, especially when the dimension is high. In online estimation problems, we have to be able to obtain the estimate $E(X_t|Y_{1:t})$ immediately after observing y_t, and that limits the amount of computation time. Because of the success of the LEnKF for high dimensional linear Gaussian SSMs, in this chapter, we develop a localized PF for general high dimensional SSMs.

3.1 The Localized Augmented Particle Filter (LAPF)

Similar to the localization procedure in the LEnKF, we consider the localization for PFs that allows us to update each block x_{t,K_j} of the state vector based on the corresponding block y_{t,N_j} of the observation vector, where the indices K_j and N_j indicate which components are in the blocks of x_t and y_t, respectively. Here we do not allow the blocks of x_t to overlap with each other, and similarly for the blocks of y_t. For the localized PF, we borrow the idea of serial updating in the EnKF and develop an algorithm that allows blockwise updating to construct the whole state vector x_t.

Note that the EnKF serial updating gives us an estimator which is the same as a non-serial updating estimator. However, We do not pursue to find the serial updating PF that provides the same estimates as the non–serial updating PF. The two methods, the localization and the serial updating, which makes the EnKF effective in the high dimensional SSMs, have different purposes. However, what they do in the EnKF looks quite similar: Both of them follow a blockwise update, that is, to update x_{t,K_j} by using the corresponding block y_{t,N_j}. In the localized EnKF, the influence of y_t on updating x_{t,K_j} can be suppressed, and only a small part of y_t would be used for the updating. In the serial updating EnKF, the conditional independence of y_t given x_t can be utilized to do the blockwise update. Hence, in this chapter, we focus on developing an algorithm that allows to do a blockwise update to construct the whole state vector x_t.

First, we have to choose the block x_{t,K_j} and the corresponding block y_{t,N_j} which will be used to update x_{t,K_j}. The block x_{t,K_j} can be chosen based on the covariance structure of the state noise, the geometrical distance between the components of x_t, or any prior knowledge about the structure of x_t. Once we decide x_{t,K_j}, the corresponding block y_{t,N_j} can be chosen by the measurement equation. It is also possible to choose the blocks of y_t first, and then find the corresponding blocks of x_t afterwards. This strategy may be useful when our knowledge about the structure of x_t is very limited. After we have the two sets of blocks, we draw particles from the local target density by considering $p(y_{t,N_j}|x_{t,K_j})$ as the local measurement density, $p(x_{t,K_j}|x_{t-1})$ as the local state density, and $p(x^f_{t,K_j}|x_{t-1})$ as the local augmented state density. The details of the localized augmented particle filter are given as follows for each time t.

1. Draw a forecast particle $x^{f,(i)}_{t,K_j}$ from $p(x^f_{t,K_j}|x^{(i)}_{t-1})$.
2. Draw a likelihood particle $x^{l,(i)}_{t,K_j}$ from $q_l(x^l_{t,K_j}|y_{t,N_j})$.
3. Combine the two particles $x^{f,(i)}_{t,K_j}$ and $x^{l,(i)}_{t,K_j}$ through a linear combination as in (2.2) to obtain $x^{(i)}_{t,K_j}$.
4. Compute the weight $w^{(i)}_{t,K_j} = \frac{p(y_{t,N_j}|x^{(i)}_{t,K_j})p(x^{(i)}_{t,K_j}|x^{(i)}_{t-1})}{q_l(x^{l,(i)}_{t,K_j}|y_{t,N_j})}$.
5. Resample the blocks $\{x^{(i)}_{t,K_j}\}^N_{i=1}$ with probability proportional to $\{w^{(i)}_{t,K_j}\}^N_{i=1}$.
6. Repeat Steps 1-5 for every block $j = 1, \ldots, M$.
7. Combine each block by match their index i to construct $x^{(i)}_t$. For each overlapping component, take the simple average of that component in different blocks.

Then, a simple average of $g(x^{(i)}_t)$ is the estimate of $E(g(X_t)|Y_{1:t})$. In the above, we want $q_l(x^l_{t,K_j}|y_{t,N_j})$ to be close to $p(y_{t,N_j}|x_{t,K_j})$ which is an approximated marginal density of y_{t,N_j} under $p(y_t|x_t)$. The updating procedure of x_t (sampling and resampling) can be done in a smaller dimension than the dimension of the original problem.

This approach can increase the quality of particles, and provide better estimates for the hidden state x_t than other approaches as shown in Section 3.3. Since now the sampling and resampling are done in a lower dimension, the localized augmented PF can avoid the problems caused by the high dimensionality in the SSM. Theocratical justification of this approach is given in the next section.

3.2 The Convergence of the LAPF

We provide a theoretical justification for the localized APF for the simplest case. Note that x_{t,K_j} denotes the j-th block of the state vector, and y_{t,N_j} is the corresponding block of the observation vector.

Assumption 0 The blocks $y_{t,N_1}, \dots, y_{t,N_M}$ and $x_{t,K_1}, \dots, x_{t,K_M}$ have no overlaps.

Assumption 1-1 The measurement density can be decomposed as

$$p(y_t|x_t) = \prod_{j=1}^{M} p(y_{t,N_j}|x_{t,K_j}),$$

which requires y_{t,N_j}'s to be conditionally independent given x_t, and each y_{t,N_j} depends only on x_{t,K_j}.

Assumption 1-2 All blocks of x_t, $x_{t,K_1}, \dots, x_{t,K_M}$ are conditionally independent given x_{t-1}. Also,

$$p(x_{t,K_j}|x_{t-1}) = p(x_{t,K_j}|x_{t-1,K_j}).$$

Assumption 2-1 For all j, $p(x_{t,K_j}|y_{1:t}) = p(x_{t,K_j}|y_{t,N_j}, y_{1:t-1})$.

Assumption 2-2 For all j, $p(x_t|y_{1:t}) = \prod_{j=1}^{M} p(x_{t,K_j}|y_{1:t})$.

At each time t, the LAPF procedure generates the weighted samples from each block independently and combines the samples for each component of x_t to construct samples for the whole state vector x_t after resampling. The final samples for x_t can be viewed as the samples from $p(x_t|y_{1:t})$. Unlike the true state vector x_t, given x_{t-1} the augmented state vector x_t^f is conditionally independent with every x_t and y_t except x_{t-1}. Hence, we use $x_{t,K_1}^f, \dots, x_{t,K_M}^f$ as the blocks for x_t^f according to the blocks of x_t in Assumption 0.

Proposition 3.2.1. *Under Assumption 0, Assumptions 1-1 and 1-2 imply Assumptions 2-1 and 2-2.*

Proof. We will prove the proposition via the mathematical induction. First, the proposition is true at $t = 1$ since for any j, we have

$$\begin{aligned}
&p(x_{1,K_j}|y_1) \\
&\propto p(y_{1,N_j}|x_{1,K_j})p(x_{1,K_j}) \\
&\propto p(y_{1,N_j}^-)p(y_{1,N_j}|x_{1,K_j})p(x_{1,K_j}) \\
&\propto p(y_{1,N_j}^-)p(x_{1,K_j}|y_{1,N_j}) \\
&\propto p(x_{1,K_j}|y_{1,N_j}),
\end{aligned}$$

where y_{1,N_j}^{-} is the components of y_t that are not in y_{1,N_j}. Also, because of Assumption 1-1 we have

$$\begin{aligned}
&p(x_1|y_1)\\
&\propto \prod\nolimits_{j=1}^{M} p(y_{1,N_j}|x_{1,K_j})p(x_{1,K_j})\\
&\propto \prod\nolimits_{j=1}^{M} p(y_{1,N_j}|x_{1,K_j})p(x_{1,K_j})\\
&\propto \prod\nolimits_{j=1}^{M} p(x_{1,K_j}|y_{1,N_j})\\
&\propto \prod\nolimits_{j=1}^{M} p(x_{1,K_j}|y_1).
\end{aligned}$$

Suppose Assumptions 2-1 and 2-2 hold at $t-1$. Let x_{t,K_j}^{-} denote components of x_t that are not in x_{t,K_j}. Then, we have

$$\begin{aligned}
&p(x_{t,K_j}|y_{1:t})\\
&\propto \int\int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}dx_{t,K_j}^{-}\\
&= \int\int \prod\nolimits_{k=1}^{M} p(y_{t,N_k}|x_{t,K_k})p(x_{t,K_k}|x_{t-1,K_k})p(x_{t-1,K_k}|y_{1:t-1}) \prod\nolimits_{k=1}^{M} dx_{t-1,K_k}dx_{t,K_j}^{-}\\
&= \int \prod\nolimits_{k\neq j} p(y_{t,N_k}|x_{t,K_k})p(x_{t,K_k}|y_{1:t-1})dx_{t,K_j}^{-}\times\\
&\int p(y_{t,N_j}|x_{t,K_j})p(x_{t,K_j}|x_{t-1,K_j})p(x_{t-1,K_j}|y_{1:t-1})dx_{t-1,K_j}\\
&\propto \int p(y_{t,N_j}|x_{t,K_j})p(x_{t,K_j}|x_{t-1,K_j})p(x_{t-1,K_j}|y_{1:t-1})dx_{t-1,K_j}\\
&= p(y_{t,N_j}|x_{t,K_j})p(x_{t,K_j}|y_{1:t-1})\\
&\propto p(x_{t,K_j}|y_{t,N_j},y_{1:t-1})
\end{aligned}$$

Also, we have

$$\begin{aligned}
&p(x_t|y_{1:t})\\
&\propto p(y_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}\\
&= \prod\nolimits_{j=1}^{M} p(y_{t,N_j}|x_{t,K_j})\int \prod\nolimits_{j=1}^{M} p(x_{t,K_j}|x_{t-1,K_j})p(x_{t-1,K_j}|y_{1:t-1})dx_{t-1}\\
&= \prod\nolimits_{j=1}^{M} p(y_{t,N_j}|x_{t,K_j})\prod\nolimits_{j=1}^{M} \int p(x_{t,K_j}|x_{t-1,K_j})p(x_{t-1,K_j}|y_{1:t-1})dx_{t-1,K_j}\\
&= \prod\nolimits_{j=1}^{M} p(y_{t,N_j}|x_{t,K_j})\prod\nolimits_{j=1}^{M} p(x_{t,K_j}|y_{1:t-1})\\
&\propto \prod\nolimits_{j=1}^{M} p(x_{t,K_j}|y_{t,N_j},y_{1:t-1}).\\
&\propto \prod\nolimits_{j=1}^{M} p(x_{t,K_j}|y_{1:t}).
\end{aligned}$$

By the mathematical induction, Assumptions 2-1 and 2-1 hold for all t.

$\square$

Theorem 3.2.2. *Under Assumptions 0, 2-1 and 2-2, the LAPF estimate which utilizes the same partition of x_t and y_t as in Assumption 0 converges to its target under $p(x_t|y_{1:t})$ as the sample size n goes to ∞.*

Proof. Suppose we have the samples $\{x_{t-1}^{(i)}\}_{i=1}^N$ from $p(x_{t-1}|y_{1:t-1})$. For each block $j = 1, \ldots, M$, we obtain the weighted samples $\{x_{t,K_j}^{(i)}, x_{t,K_j}^{f,(i)}, x_{t-1}^{(i)}, w_{t,K_j}^{(i)}\}_{i=1}^N$, whose target density is

$$\begin{aligned}
& p(x_{t,K_j}, x_{t,K_j}^f, x_{t-1}|y_{t,N_j}, y_{1:t-1}) \\
& \propto p(y_{t,N_j}, x_{t,K_j}, x_{t,K_j}^f, x_{t-1}|y_{1:t-1}) \\
& = p(y_{t,N_j}|x_{t,K_j})p(x_{t,K_j}, x_{t,K_j}^f, x_{t-1}|y_{1:t-1}) \\
& = p(y_{t,N_j}|x_{t,K_j})p(x_{t,K_j}|x_{t-1})p(x_{t,K_j}^f|x_{t-1})p(x_{t-1}|y_{1:t-1}).
\end{aligned}$$

Also, the joint proposal density for $\{x_{t,K_j}^{(i)}, x_{t,K_j}^{f,(i)}, x_{t-1}^{(i)}, w_{t,K_j}^{(i)}\}_{i=1}^N$ is

$$q(x_{t,K_j}, x_{t,K_j}^f|y_{t,N_j}, x_{t-1})q(x_{t-1}|y_{1:t-1}) \propto q_l(x_{t,K_j}^l|y_{t,N_j})p(x_{t,K_j}^f|x_{t-1})p(x_{t-1}|y_{1:t-1}).$$

Thus, the weight for the set of particles $\{x_{t,K_j}^{(i)}, x_{t,K_j}^{f,(i)}, x_{t-1}^{(i)}\}$ is given to be

$$w_{t,K_j}^{(i)} = \frac{p(y_{t,N_j}|x_{t,K_j}^{(i)})p(x_{t,K_j}^{(i)}|x_{t-1}^{(i)})}{q_l(x_{t,K_j}^{l,(i)}|y_{t,N_j})}.$$

The detailed computation of the weight is as follows.

$$\begin{aligned}
& \frac{p(y_{t,N_j}|x_{t,K_j}^{(i)})p(x_{t,K_j}^{(i)}|x_{t-1}^{(i)})p(x_{t,K_j}^{f,(i)}|x_{t-1}^{(i)})p(x_{t-1}^{(i)}|y_{1:t-1})}{q(x_{t,K_j}^{(i)}, x_{t,K_j}^{f,(i)}|y_{t,N_j}, x_{t-1}^{(i)})p(x_{t-1}^{(i)}|y_{1:t-1})} \\
= & \frac{p(y_{t,N_j}|x_{t,K_j}^{(i)})p(x_{t,K_j}^{(i)}|x_{t-1}^{(i)})p(x_{t,K_j}^{f,(i)}|x_{t-1}^{(i)})}{q_l(x_{t,K_j}^{l,(i)}|y_{t,N_j})p(x_{t,K_j}^{f,(i)}|x_{t-1}^{(i)})} \\
= & \frac{p(y_{t,N_j}|x_{t,K_j}^{(i)})p(x_{t,K_j}^{(i)}|x_{t-1}^{(i)})}{q_l(x_{t,K_j}^{l,(i)}|y_{t,N_j})}.
\end{aligned}$$

After the blockwise resampling, $\{x_{t,K_j}^{(i)}\}_{i=1}^N$ alone, without the particles for the augmented space x_t^f and the particles for the previous time step $t-1$, can be treated as the samples from $p(x_{t,K_j}|y_{t,N_j}, y_{1:t-1}) = p(x_{t,K_j}|y_{1:t})$. In our LAPF implementation, the particles for each block are generated independently. The history of different blocks does not affect the sampling of the current block j, so $\{x_{t,K_j}^{(i)}\}_{i=1}^N$ are independent to any other particles for different blocks. At the end, we combine particles through matching index i to construct the final samples from $p(x_t|y_{1:t})$. Notice that $p(x_t|y_{1:t}) = \prod_{j=1}^{M} p(x_{t,K_j}|y_{1:t})$, so the final particles are the samples from $p(x_t|y_{1:t})$. □

A SSM that satisfies Assumptions 1-1 and 1-2, can be found in linear Gaussian models.

$$\begin{cases} y_t|x_t = x_t + u_t \\ x_t|x_{t-1} = x_{t-1} + v_t \end{cases}$$

where $u_t \sim N(0, \mathbf{I})$ and $v_t \sim N(0, \mathbf{I})$. Under this model, any choice of blocks in x_t can satisfy the two assumptions. The model can be relaxed with block diagonal covariance matrices with the same shape for measurement and state equations, but in this case blocks of x_t must be chosen to match the blocks in the covariance matrices.

3.3 Simulation Studies

3.3.1 Lorenz-96 Model

We revisit the Lorenz-96 model in Section 2.4.4. The Lorenz-96 model is defined by a set of differential equations over continuous time ϕ and discrete space $i = 1, \dots, k$:

$$\frac{\xi_{\phi,i}}{d\phi} = f_i(\phi, \xi_\phi) = (\xi_{\phi,i+1} - \xi_{\phi,i-2})\xi_{\phi,i-1} - \xi_{\phi,i} + F.$$

Here $k = 100$ is the dimension of ξ_ϕ and the constant $F = 8$. To discretize the given differential equations, we implemented the fourth order Runge-Kutta method with time step $\Delta\phi = 0.05$. That is,

$$\xi_{\phi+\Delta\phi} = \xi_\phi + \frac{1}{6}\Delta\phi(k_1 + 2k_2 + 2k_3 + k_4), \tag{3.1}$$

where

$$\begin{aligned} k_1 &= f(\phi, \xi_\phi), \\ k_2 &= f(\phi + \frac{1}{2}\Delta\phi, \xi_\phi + \frac{1}{2}\Delta\phi \cdot k_1), \\ k_3 &= f(\phi + \frac{1}{2}\Delta\phi, \xi_\phi + \frac{1}{2}\Delta\phi \cdot k_2), \\ k_4 &= f(\phi + \Delta\phi, \xi_\phi + \Delta\phi \cdot k_3). \end{aligned}$$

Recall that we define $\xi_{\phi,0} := \xi_{\phi,k}$, $\xi_{\phi,-1} := \xi_{\phi,k-1}$, and $\xi_{\phi,k+1} := \xi_{\phi,1}$. For notational convenience, let $x_{t+1,i}$

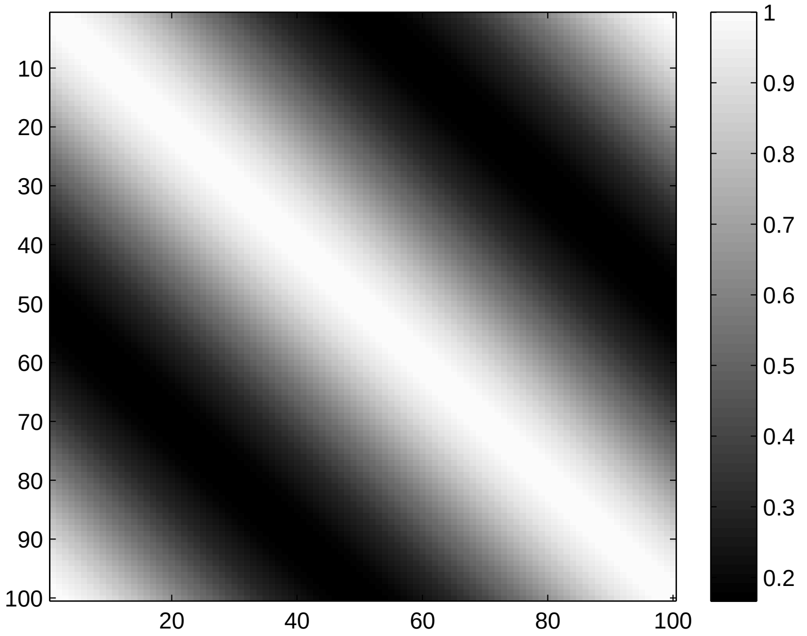

Figure 3.1: A graphical representation of the covariance matrix Q. Each component of Q is converted to the gray scale with white being 0 and black being 1.

denote $\xi_{t\Delta\phi,i}$. Then, we have the discretized nonlinear state equation after adding random perturbations to (3.1) as follows.

$$x_t = x_{t-1} + \frac{1}{6}\Delta\phi(k_1 + 2k_2 + 2k_3 + k_4) + v_t, \tag{3.2}$$

where v_t is Gaussian noise with mean 0 and a banded covariance matrix Q, which is shown in Figure 3.1. Each (i, j)-th component of Q is related to the distance between the i-th and j-th components of x_t. The measurement equation is given as

$$y_t = x_t + u_t,$$

where $u_t \sim N(0, \mathbf{I}_{100})$.

In the following simulation study, we compared five methods: the localized augmented particle filter (LAPF), the optimal Kalman filter (OPF), the localized ensemble Kalman filter (LEnKF), the independent particle filter (IPF), and the nonlinear ensemble adjustment filter (NLEAF) proposed by Lei and Bickel (2009). The NLEAF incorporates the EnKF and the NPF, and it can be implemented with the localization. However, their localized estimate is not generally consistent. For the LAPF and the NLEAF, we use the notation $C, \{b, a\}$ to denote the setting with C blocks and b and a denote the number of components behind and ahead of the center of each block. The centers of the blocks for a fixed C are chosen to be

Table 3.1: The comparison of RMSE and its standard error for four methods for Lorenz 96 model with $g(x) = x$.

		$RMSE$	$se(RMSE)$
OPF		54.681187	0.535504
IPF		13.385649	0.010565
LAPF	$C = 100, \{0, 0\}$	8.483723	0.011024
NLEAF	$C = 100, \{0, 0\}$	8.544413	0.014321

Table 3.2: The comparison of RMSE and its standard error for four methods for Lorenz 96 model with $g(x) = \exp(x/||x||^2)$.

		$RMSE$	$se(RMSE)$
OPF		0.023341	0.000216
IPF		0.009284	0.000254
LAPF	$C = 100, \{0, 0\}$	0.006626	0.000219
NLEAF	$C = 100, \{0, 0\}$	0.006708	0.000219

$\{\frac{k}{C}(j-1)+1\}_{j=1}^{C}$. With $K = 100$ (number of experiments) and $T = 200$ (number of time steps), we define the root mean squared error and its standard error as the following:

$$\text{RMSE} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{T}||\hat{g}(X_{1:T}^{k}) - g(X_{1:T}^{k})||^2}$$

$$\text{se(RMSE)} = \sqrt{\frac{1}{K}\widehat{Var}\left(\sqrt{\frac{1}{T}||\hat{g}(X_{1:T}^{k}) - g(X_{1:T}^{k})||^2}\right)}$$

Note that we consider two different cases for $g(\cdot)$: 1) $g(x) = x$ and 2) $g(x) = \exp(x/||x||^2)$, where in $\exp(x)$ we apply $\exp(\cdot)$ to each component of x. Also, in both OPF and IPF we need to do resampling at every step t; otherwise the estimate would diverge.

In Table 3.1 and 3.2, we can see that the LAPF does much better than the OPF and the IPF. It is interesting to see that the LAPF outperforms the OPF with a significant margin. This indicates that the PF should be implemented with localization in high dimensional SSMs.

Chapter 4

Particle Filtering with Independent Batches

In this chapter, we discuss the improvements of the PF by dividing the particles into independent batches. The development of the method is motivated by the particle Markov chain Monte Carlo method proposed by Andrieu et al. (2010), so we will borrow the new notations from their paper, which are given in the following section.

4.1 Review of the PF

In this chapter, the state space model has minor changes. Unlike the one we define in Section 1.1, the sequence of the state vector begins with time step $n = 1$. Also, the initial state vector X_1 has a observation Y_1, as illustrated in Figure 4.1. The model can be presented by

$$\begin{cases} X_1 \sim \mu_\theta(\cdot) \\ (X_{n+1}|X_n = x) \sim f_\theta(\cdot|x) \\ (Y_n|X_n = x) \sim g_\theta(\cdot|x). \end{cases} \tag{4.1}$$

The PF with resampling at every time step is presented below. In Step 2 a), A^k_{n-1} denotes the index of the resampled particle at time step $n-1$. A distribution function F describes the resampling procedure we use with probability proportional to $\mathbf{W_{n-1}} = (W^1_{n-1}, \ldots, W^k_{n-1}, \ldots, W^N_{n-1})$.

Step 1: At time $n = 1$,

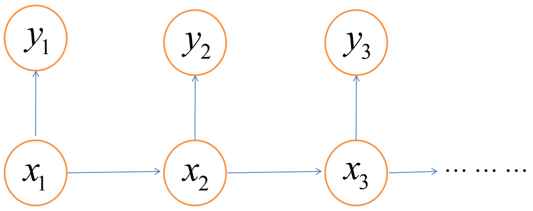

Figure 4.1: The illustration of the state space model in Chapter 4.

(a) sample $X_1^k \sim q_\theta(\cdot|y_1)$ and

(b) compute and normalize the weights

$$w_1(X_1^k) := \frac{\mu_\theta(X_1^k)g_\theta(y_1|X_1^k)}{q_\theta(X_1^k|y_1)}$$
$$W_1^k := \frac{w_1(X_1^k)}{\sum\limits_{m=1}^{N} w_1(X_1^m)}$$

Step 2: At times $n = 2, \ldots, T$

(a) sample $A_{n-1}^k \sim F(\cdot|\mathbf{W_{n-1}})$

(b) sample $X_n^k \sim q_\theta(\cdot|y_n, X_{n-1}^{A_{n-1}^k})$ and set $X_{1:n}^k := (X_{1:n-1}^{A_{n-1}^k}, X_n^k)$, and

(c) compute and normalize the weights

$$w_n(X_{1:n}^k) := \frac{f_\theta(X_n^k|X_{n-1}^{A_{n-1}^k})g_\theta(y_n|X_n^k)}{q_\theta(X_n^k|y_n, X_{n-1}^{A_{n-1}^k})},$$
$$W_n^k := \frac{w_n(X_{1:n}^k)}{\sum\limits_{m=1}^{N} w_n(X_{1:n}^m)}.$$

Then, we have

$$\sum\nolimits_{k=1}^{N} W_n^k g(X_{1:T}^k) \xrightarrow{p} E(g(X_{1:T})|Y_{1:T}) \text{ as } N \to \infty.$$

We make use of the notion of ancestral lineage $B_{1:T}^k = (B_1^k, B_2^k, ..., B_{T-1}^k, B_T^k = k)$ of a path $X_{1:T}^k = (X_1^{B_1^k}, X_2^{B_2^k}, ...X_{T-1}^{B_{T-1}^k}, X_T^{B_T^k})$ where $B_T^k := k$ and $B_n^k = A_n^{B_{n+1}^k}$ for $n = T-1, ..., 1$. The ancestral lineage $B_{1:T}^k$ allows us to track down all ancestors of the k-th particle X_T^k, so we can more easily handle the randomness of the resampling procedure in the proof of theocratical results. See Andrieu et al. (2010) for more details of the notations.

4.2 Independent Batches for the PF

Andrieu et al. (2010) proposed the particle Markov chain Monte Carlo method (PMCMC) which utilizes the proposal density constructed from the particle filter. Their main interest is to solve the off–line estimation problem, and the PMCMC might not be suitable for the on–line filtering problem. Our goal is to utilize the results from the PMCMC to improve the quality of the PF estimate.

Let us assume that we can afford to generate NL particles to run the SIR algorithm in Section 4.1. We find that running L independent SIR batches with N particles each, and combining L different estimates in a certain way can outperform running the ordinary SIR with NL particles. So this could be a more efficient way to implement the PF.

Let us consider running SIR in L independent batches. In each batch, we perform SIR with N samples, and compute

$$\hat{Z}^N := \prod_{n=1}^{T} \{\frac{1}{N} \sum_{k=1}^{N} w_n(X_{1:n}^k)\}.$$

Note that $\hat{Z}^N$ is an estimate of the marginal likelihood $p(y_{1:T})$. Let the index i denote the batch number. After carrying out SIR at all batches, we need to combine the estimates from each batch as the following:

$$\hat{\mu} := \sum_{i=1}^{L} \tilde{z}(i) \sum_{k=1}^{N} W_P^k(i) g(X_{1:P}^k(i)), \text{ where } \tilde{z}(i) = \frac{\hat{Z}^N(i)}{\sum\limits_{i=1}^{L} \hat{Z}^N(i)}. \tag{4.2}$$

We call this method the sequential importance resampling within batches (SIRB). The given estimate converges to $E(g(X_{1:T})|Y_{1:T})$ as either N or L goes to infinity, and the proof of the convergence is given in Section 4.3. The convergence as N goes infinity is trivial, but the result as L goes to infinity is rather interesting, since it implies that increasing the number of batches with a fixed number of particles within each batch guarantees the convergence of $\hat{\mu}$.

Let $\mu = E(g(X_{1:T})|Y_{1:T})$. The asymptotic mean and variance of the estimate $\hat{\mu}$ is given by the Taylor series approximation from Givens and Hoeting (2005):

$$E(\hat{\mu}) = \mu - \frac{1}{NL} \operatorname{cov}(X_{1:T}, \tilde{z}W_T) + \frac{\mu}{NL} \operatorname{var}(\tilde{z}W_T) + O(\frac{1}{(NL)^2}),$$
$$\operatorname{var}(\hat{\mu}) = \frac{1}{NL}(\operatorname{var}(X_{1:T}) + \mu \operatorname{var}(\tilde{z}W_T)\mu' - 2\mu \operatorname{cov}(X_{1:T}, \tilde{z}W_T) + O(\frac{1}{(NL)^2}).$$

The variance and covariance in the expressions above are taken with respect to all the random variables generated in the SIR algorithm. We know that $\hat{\mu}$ converges to μ as either N or L goes to infinity. However, we can see that increasing N might reduce the mean square error of the estimate faster than increasing L because of the following reason. Notice that the SIR estimate $\hat{Z}^N$ converges to $p(y_{1:T})$. So, we have $\operatorname{var}(\tilde{z}W_T)$ goes to zero as N goes to infinity, but increasing L does not affect $\operatorname{var}(\tilde{z}W_T)$ much. Therefore, it is better to have a small number of batches with a large sample size in each batch than to have a large number of batches with a small sample size in each batch.

4.3 Convergence of the SIRB

All the notation and equation numbers follow those in Andrieu et al. (2010) unless defined otherwise. With N particles and L batches, the SIRB estimate of $E(g(X_{1:P})|Y_{1:P})$ from the model 4.1 is defined by

$$\sum_{i=1}^{L}\tilde{z}^*(i)\sum_{k=1}^{N}W_P^{*k}(i)g(X_{1:P}^{*k}(i)) \text{ where } \tilde{z}^*(i)=\frac{\hat{Z}^{N,*}(i)}{\sum\limits_{i=1}^{L}\hat{Z}^{N,*}(i)}$$

The index i denote the i-th SIR batch. The superscript star can be ignored in our setting, but we keep the stars to have the consistent notation with Andrieu et al. (2010). Under Assumption 1 and Assumption 2 in Andrieu et al. (2010), for any $N \geq 1$ the SIRB estimate converges to $E(g(X_{1:P})|Y_{1:P})$ as L goes to infinity. Let's consider a proposal distribution q_k^N to describe the particle generating procedure and the resampling procedure with a realization of a set of random variables $\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1}$ as the SIR in a single batch except at the last step P. To use the results in Andrieu et al. (2010), we make q_k^N to deterministically choose the k-th particle with all ancestors, $x_{1:P}^k = (x_1^{b_1^k}, x_2^{b_2^k}, ..., x_{P-1}^{b_{P-1}^k}, x_P^{b_P^k})$ where b_p^k is a realization of the ancestral lineage B_p^k. Thus, q_k^N can be written as:

$$q_k^N(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1}) := \psi(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1}),$$

where

$$\psi(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1}) := \left\{\prod_{m=1}^{N}M_1(x_1^m)\right\}\prod_{n=2}^{P}\left\{r(\mathbf{a}_{n-1}|\mathbf{w}_{n-1})M_n(x_n^m|x_{1:n-1}^{a_{n-1}^m})\right\}.$$

With the density $r(\cdot)$ for the resampling procedure with probabilities proportional to $\mathbf{w}_{n-1}$ and the proposal density $M_n(\cdot)$ to draw X_n, $\psi(\cdot)$ describes the density of all the random variables (particles and random indexes after resampling) generated by the SIR. Also, our extended target $\tilde{\pi}_k^N$ is similar to the density in equation (31) defined in Andrieu et al. (2010) except the fact that we fixed k:

$$\tilde{\pi}_k^N(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1}) := \frac{\pi(x_{1:P}^k)}{N^P}\frac{\psi(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1})}{M_1(x_1^{b_1^k})\prod\limits_{n=2}^{P}r(b_{n-1}^k|\mathbf{w}_{n-1})M_n(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k})}.$$

Note the following fact from Andrieu et al. (2010):

$$\int g(x_{1:P}^k)\frac{\pi(x_{1:P}^k)}{N^P}\frac{\psi(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1})}{M_1(x_1^{b_1^k})\prod\limits_{n=2}^{P}r(b_{n-1}^k|\mathbf{w}_{n-1})M_n(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k})}d(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_P,\mathbf{a}_1,\ldots,\mathbf{a}_{P-1})$$

$$= Z \cdot E(g(X_{1:P})|Y_{1:P}),$$

which implies if we fix k in the integrand and do the integral with respect to $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}$ in the above, the results would be $\frac{Z}{N} E(g(X^k_{1:P})|Y_{1:P})$.

Assume we choose the k-th path $x^k_{1:P}$ from every SIR batch. Those sets of particles can be viewed as samples from the proposal density q^N_k. With the fact in (41) of Andrieu et al. (2010):

$$\hat{Z}^N(i) W^k_P(i) q^N_k(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) = Z \tilde{\pi}^N_k(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}),$$

we have as L goes to infinity,

$$\begin{aligned}
&\frac{1}{L}\sum_{i=1}^{L} \hat{Z}^{N,*}(i) W^{*k}_P(i) g(X^{*k}_{1:P}(i)) \\
\xrightarrow{p}\ & \textstyle\int g(x^k_{1:P}) \hat{Z}^N(i) W^k_P(i) q^N_k(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) d(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) \\
&= Z \textstyle\int g(x^k_{1:P}) \tilde{\pi}^N_k(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) d(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) \\
&= Z \textstyle\int g(x^k_{1:P}) \frac{\pi(x^k_{1:P})}{N^P} \frac{\psi(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1})}{M_1(x_1^{b^k_1}) \prod\limits_{n=2}^{P} r(b^k_{n-1}|\mathbf{w}_{n-1}) M_n(x_n^{b^k_n}|x_{1:n-1}^{b^k_{n-1}})} d(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_P, \mathbf{a}_1, \ldots, \mathbf{a}_{P-1}) \\
&= \frac{Z}{N} E(g(X_{1:P})|Y_{1:P}).
\end{aligned}$$

Also, $\sum\limits_{i=1}^{L} \hat{Z}^{N,*}(i)$ is a consistent estimate of Z in terms of L. Therefore, we have

$$\begin{aligned}
\sum_{i=1}^{L} \tilde{z}^*(i) \sum_{k=1}^{N} W^{*k}_P(i) g(X^{*k}_{1:P}(i)) \quad &= \quad \frac{\frac{1}{L}\sum\limits_{i=1}^{L} \hat{Z}^{N,*}(i) \sum\limits_{k=1}^{N} W^{*k}_P(i) g(X^{*k}_{1:P}(i))}{\frac{1}{L}\sum\limits_{i=1}^{L} \hat{Z}^{N,*}(i)} \\
&\xrightarrow{p} \quad \frac{Z \cdot E(g(X^k_{1:P})|Y_{1:P})}{Z}, \ L \to \infty \\
&= \quad E(g(X^k_{1:P})|Y_{1:P}).
\end{aligned}$$

4.4 Simulation Studies

In this section we compare the SIR and the SIR with batches (SIRB). We want to show that SIR with batches works better with large N rather than large L. The following model will be used to test the algorithms.

$$\begin{cases} Y_n|X_n = X_n^2/20 + W_n \\ X_n|X_{n-1} = 0.5X_{n-1} + \frac{25X_{n-1}}{1+X_{n-1}^2} + 8\cos(1.2n) + V_n, \end{cases}$$

where W_n and V_n are Gaussian noise terms with variance σ_V^2 and σ_W^2. Also, $X_1 \sim N(0, 5)$. With $\sigma_V^2 = 1$ and $\sigma_W^2 = 10$, we generated 300 observations $y_{1:300}$. The root mean square error is used as the performance criteria:

$$\text{RMSE} = \frac{1}{K} \sum_{j=1}^{K} \sqrt{\frac{1}{T} ||\hat{X}_{1:T,(j)} - X_{1:T,(j)}||^2},$$

where K is the number of independent experiments, $X_{1:T,(j)}$ is a true state vector for the j-th experiment, and $\hat{X}_{1:T,(j)}$ is the estimate. Let $K = 100$. The RMSE is evaluated for six values of $T = (50, 100, 150, 200, 250, 300)$ to compare the performances for different length of the time series. For SIR with batches, it is equivalent to draw NL samples. We fix $N^* = N \cdot L$, and consider five different settings:

SMC N=1,000,000.

Bat1 N=25 and L=40,000.

Bat2 N=100 and L=10,000.

Bat3 N=250 and L=4,000.

Bat4 N=1,000 and L=1,000.

From the simulation results, we can see that for fixed $N^* = NL$, increasing N is more effective in reducing the RMSE. Also, the five methods compared in the study are not significantly different at small T, but the difference becomes significant at large T. Not all the SIRBs work better than SIR. With the first setting, $N = 25$ and $L = 40,000$, SIR with batches is actually doing worse than SIR. However, in the other three SIR settings with batches did better than the SIR for large T, and this seems to suggest that they may do well in high dimensional problems.

Table 4.1: A Comparison of the average RMSE and standard errors for the SIR and the SIR with batches based on $K = 100$ repeated experiments. SMC: N=1,000,000, Bat1: N=25 and L=40,000, Bat2: N=100 and L=10,000, Bat3: N=250 and L=4,000, and Bat4: N=1,000 and L=1,000.

	T=10		T=25	
	RMSE	se(RMSE)	RMSE	se(RMSE)
SMC	2.962	0.426	2.946	0.255
*Bat*1	2.364	0.254	2.460	0.185
*Bat*2	2.414	0.254	2.399	0.142
*Bat*3	2.57	0.327	2.586	0.198
*Bat*4	2.542	0.323	2.63	0.201
	T=50		T=100	
	RMSE	se(RMSE)	RMSE	se(RMSE)
SMC	2.364	0.172	2.096	0.119
*Bat*1	2.206	0.129	2.375	0.100
*Bat*2	2.054	0.099	1.980	0.085
*Bat*3	2.139	0.132	1.941	0.095
*Bat*4	2.129	0.130	1.931	0.090
	T=150		T=200	
	RMSE	se(RMSE)	RMSE	se(RMSE)
SMC	2.174	0.097	2.245	0.079
*Bat*1	2.479	0.081	2.584	0.073
*Bat*2	2.006	0.079	2.094	0.066
*Bat*3	1.923	0.078	1.983	0.067
*Bat*4	1.934	0.076	1.937	0.064
	T=250		T=300	
	RMSE	se(RMSE)	RMSE	se(RMSE)
SMC	2.273	0.069	2.297	0.064
*Bat*1	2.741	0.069	2.770	0.073
*Bat*2	2.188	0.061	2.150	0.056
*Bat*3	2.027	0.062	2.024	0.055
*Bat*4	1.946	0.058	1.924	0.054

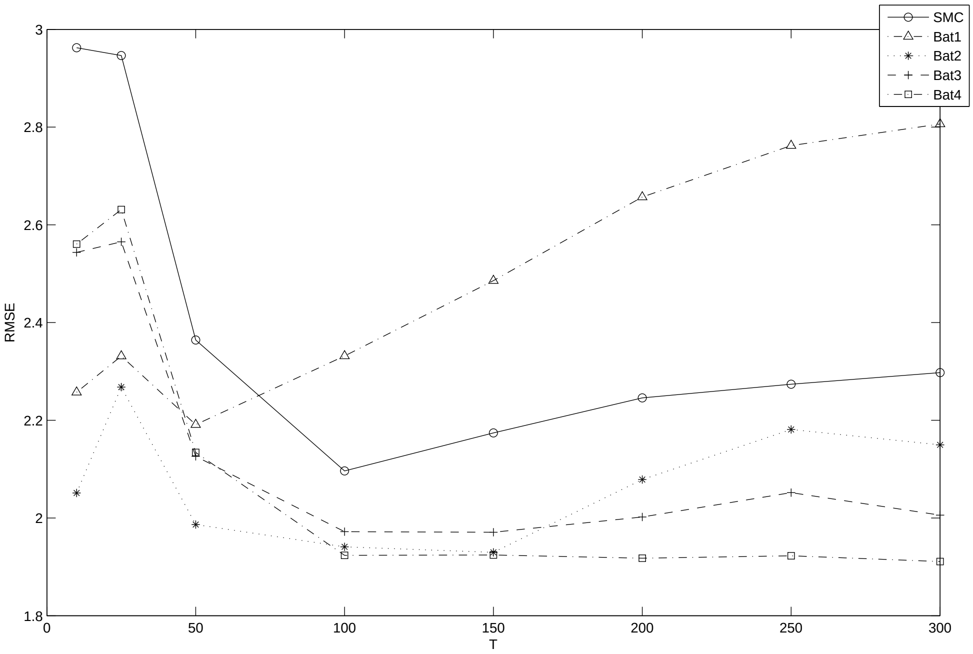

Figure 4.2: Comparison of the average RMSE for the SIR and the SIR with batches based on $K = 100$ repeated experiments. SMC: N=1,000,000, Bat1: N=25 and L=40,000, Bat2: N=100 and L=10,000, Bat3: N=250 and L=4,000, and Bat4: N=1,000 and L=1,000.

Chapter 5

Future Work

5.1 The Generalized LAPF (GLAPF)

For the high dimensional SSMs, this is the case that the observations y_{t,N_j}'s arrives sequentially. In such case, users may want to update the posterior distribution given the components of y_t that are available, rather than waiting for all components of y_t. A new method can be utilized to sample particles from the posterior density $p(x_t|y_{t,N_1},\ldots,y_{t,N_k},y_{1:t-1})$. Also, the method utilizes the localization idea from EnKFs through the covariance tapering.

Assumption y_{t,N_j}'s are conditionally independent given x_t and each y_{t,N_j} depends only on x_{t,K_j}. When the j–th observation arrives, the target posterior density would be $p(x_t|y_{t,K_1},\ldots,y_{t,K_j},y_{1:t-1})$, and the assumption allows to decompose the density as follows:

$$\begin{aligned}
&p(x_t|y_{t,K_1},\ldots,y_{t,K_j},y_{1:t-1})\\
&\propto p(y_{t,K_1},\ldots,y_{t,K_j}|x_t)p(x_t|y_{1:t-1})\\
&\propto p(y_{t,K_j}|x_t)p(y_{t,K_1},\ldots,y_{t,K_{j-1}}|x_t)p(x_t|y_{1:t-1})\\
&\propto p(y_{t,K_j}|x_t)p(x_t|y_{t,K_1},\ldots,y_{t,K_{j-1}},y_{1:t-1})
\end{aligned}$$

To take an advantage of the given decomposition, we apply the APF for each block by treating $p(y_{t,K_j}|x_t)$ as a likelihood and $p(x_t|y_{t,K_1},\ldots,y_{t,K_{j-1}},y_{1:t-1})$ as a prior. The detail algorithm is the following:

At $t-1$, we have a set of particles $x_{t-1}^{(i)}$ from $p(x_{t-1}|y_{1:t-1})$. For $j=1$,

1. Evolve $x_{t-1}^{(i)}$ through the state equation to obtain $x_t^{f,(i)}$.

2. Obtain $\hat{p}(x_t|y_{1:t-1})$ that is an approximation of $p(x_t|y_{1:t-1})$ by using particles $x_t^{f,(i)}$.

3. Generate $x_{t,K_1}^{l,(i)}$ samples from a proposal density $q_l(x_{t,K_1}|y_{t,N_1})$.

4. Obtain weighted samples $x_{t,1}^{(i)}$ from $p(x_t|y_{t,N_1},y_{1:t-1})$ by combining $x_t^{f,(i)}$ and $x_{t,K_1}^{l,(i)}$.

5. Compute the importance weights

$$w_{t,1}^{(i)} = \frac{p(y_{t,N_1}|x_t^{1,(i)})\hat{p}(x_t^{1,(i)}|y_{1:t-1})}{q_l(x_{t,K_1}^{l,(i)}|y_{t,N_1})}$$

6. Do resampling with prob. proportional to $w_{t,1}$'s.

For $j > 1$,

1. Generate $x_{t,K_j}^{l,(i)}$ samples from a proposal density $q_l(x_{t,K_j}|y_{t,N_j})$.
2. Obtain weighted samples $x_{t,j}^{(i)}$ from $p(x_t|y_{t,K_1},\ldots,y_{t,K_j},y_{1:t-1})$ by combining $x_t^{j-1,(i)}$ and $x_{t,K_j}^{l,(i)}$.
3. Compute the importance weights

$$w_{t,j} = \frac{\prod_{k=1}^{j} p(y_{t,N_k}|x_t^{j,(i)})\hat{p}(x_t^{j,(i)}|y_{1:t-1})}{q_l(x_{t,K_j}^{l,(i)}|y_{t,N_j})}$$

4. Do resampling with probability proportional to $w_{t,j}$'s.

We deliver a few remarks:

- At $j = 1$, we construct the final particles for K_1 block by combining $x_t^{f,(i)}$ and $x_{t,K_1}^{l,(i)}$ through linear combination, and we set particles for other components to be those for $x_t^{f,(i)}$.
- Also, for the $j > 1$, we update the components in K_j block by combining particles from previous update and $x_{t,K_j}^{l,(i)}$, and the other components would be set to be the same as the particles from the previous update.
- For the density approximation, we can apply either a single normal distribution or a mixture normal distribution. Each mean and covariance matrix will be estimated by associated particles. Also, for the covariance estimation, we can consider a tapered estimate. That is,

$$C \circ \frac{1}{L}(\xi^{(i)} - \bar{\xi})(\xi^{(i)} - \bar{\xi})',$$

  where $\circ$ is a component–wise matrix product. This can be helpful to reduce a variance of the estimate when the given SSM is high–dimensional.

- We can pursue another type of density approximation:

$$\hat{p}(x_{t-1}|y_{1:t-1}) = \frac{1}{\sum w_{t-1}^{(i)}} \sum w_{t-1}^{(i)} p(x_t|x_{t-1}^{(i)}),$$

$$\hat{p}(x_t|y_{t,K_1},\ldots,y_{t,K_j},y_{1:t-1}) = C \cdot p(y_{t,K_1},\ldots,y_{t,K_j}|x_t) \frac{1}{\sum w_{t-1}^{(i)}} \sum w_{t-1} p(x_t|x_{t-1}^{(i)})$$

  where $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}$ is the weighted sample from $p(x_{t-1}|y_{1:t-1})$ and $C^{-1} = p(y_{t,K_1},\ldots,y_{t,K_j}|y_{1:t-1})$.

The implementation of the APF within each block requires to identify observation and state equations. The observation equation would be $p(y_{t,N_j}|x_t)$, but the state equation would not be given except a block with $j = 1$. This will make it hard to determine the coefficient matrices when we combine two sets of particles from $p(y_{t,N_j}|x_t)$ and $p(x_t|y_{t,K_1},\ldots,y_{t,K_{j-1}},y_{1:t-1})$. We present a way to sequentially determine the coefficient matrices. Recall the typical linear combination in the APF without localization:

$$(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1}(\hat{\Sigma}_t^{-1} x_t^{l,(i)} + \hat{Q}_t^{-1} x_t^{f,(i)})$$

In the GLAPF, we apply this update formula for the first block. That is, components of x_t which is not in x_{t,K_1} would be the same as the components of x_t^f (updated by the state equation), and the x_{t,K_1} would be updated by

$$x_{t,K_1}^{(i)} = (\hat{\Sigma}_{t,K_1}^{-1} + \hat{Q}_{t,K_1}^{-1})^{-1}(\hat{\Sigma}_{t,K_1}^{-1} x_{t,K_1}^{l,(i)} + \hat{Q}_{t,K_1}^{-1} x_{t,K_1}^{f,(i)}),$$

where $\hat{\Sigma}_{t,K_1}$ and $\hat{Q}_{t,K_1}$ are the sub–matrices of $\hat{\Sigma}_t$ and $\hat{Q}_t$ associated with x_{t,K_1}. From the second block update, we repeat the following procedure for the update.

1. Components of $x_t^{(i)}$ which is not in x_{t,K_j} would be the same as the components of $x_t^{j-1,(i)}$
2. $x_{t,K_j} = (\hat{\Sigma}_{t,K_j}^{-1} + \hat{Q}_{t,j-1,K_j}^{-1})^{-1}(\hat{\Sigma}_{t,K_j}^{-1} x_{t,K_j}^{l,(i)} + \hat{Q}_{t,K_j}^{-1} x_{t,K_j}^{j-1,(i)})$

Sampling particles in the GLAPF occurs in the lower dimension then the dimension of the given SSM, but its weight computation and resampling must be done in the full dimension in the SSM. The Performance of the GLPAF is comparable to that of the LAPF, and we want to investigate how approximations to the prediction posterior densities affect the convergence of the GLAPF.

5.2 The EnKF as a Proposal Distribution

The EnKF does not converge to $E(X_t|Y_{1:t})$ when the state space model is nonlinear or non–Gaussian. We hope to modify the EnKF to make it work for nonlinear or non–Gaussian model. As a beginning, we assume the SSM has a linear Gaussian measurement equation, but we do not make any specific assumption on the state equation.

$$\begin{cases} y_t|x_t = Hx_t + u_t, u_t \sim N(0,R) \\ x_t|x_{t-1} = f(x_{t-1}, v_t). \end{cases} \tag{5.1}$$

We want to compute the weight for the EnKF to correct its bias. For that matter, we need to compute the target density $p(x_t|y_{1:t-1})$ and the EnKF proposal density $q(x_t|y_{1:t-1})$. In the general SSM, there is no closed form expression for the two densities. However, we can approximate $p(x_t|y_{1:t-1})$ by $\hat{p}(x_t|y_{1:t-1}) = \sum_i p(x_t|x_{t-1}^{(i)})$ where $x_{t-1}^{(i)}$'s are sampled from $p(x_{t-1}|y_{1:t-1})$. Because of the high computational complexity, we can only use a few randomly selected particles in the approximation. If we can establish an one-one mapping between the forecast and analysis particles in the EnKF, we can approximate the EnKF proposal density $q(x_t|y_{1:t})$ by

$$\hat{q}(x_t|y_{1:t}) = \sum\nolimits_i \hat{p}([x_t - \hat{K}(y_t + {u_t}^{(i)})] \cdot [\mathbf{I} - \hat{K}H]^{-1}]),$$

where $\hat{K}$ is the estimated Kalman gain matrix and ${u_t}^{(i)}$'s are independent $N(0,R)$ random variables.

Then, we need to evaluate the weights as $w_t^{(i)} = \frac{p(y_t|x_t^{(i)})\hat{p}(x_t^{(i)}|y_{1:t-1})}{\hat{q}(x_t^{(i)}|y_{1:t})}$. Note that in the EnKF we do not perform any resampling. However, the resampling procedure is required in this approach since the bias correction needs to be done by modifying the ensemble generating procedure in the EnKF.

5.2.1 Gaussian Mixture Approximation for the Posterior Density

In the precious section, we pursue to correct bias of the EnKF in nonlinear or non–Gaussian models by introducing the weights. Under the same model assumption in (5.1), Bengtsson et al. (2003) proposed another way to deal with this issue by approximating the posterior densities by the mixture normal density to implement the EnKF. The method is called the ensemble mixture Kalman filter. The main weakness of their method is the approximation to the mixture normal needs to be done by a data driven method, so it may not work well in high dimensional SSMs. Hence, we want to find a way to approximate $p(x_{t+1}|y_{1:t})$ as Gaussian mixture in a more effective way. We assume that $p(x_t|y_{1:t-1})$ and $p(x_t|y_{1:t})$ at any t are Gaussian mixtures, and the number of mixtures for each density can be different. Let $p(x_t|y_t) = \sum_i \alpha_i N(x_t|\mu_i, \Sigma_i)$. Then, we have to know how to approximate the following by the mixture of Gaussian so that we can implement the mixture Kalman filter.

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t) \sum\nolimits_i \alpha_i N(x_t|\mu_i, \Sigma_i) dx_t \approx \sum\nolimits_k \beta_{ik} N(x_t|\mu_k, \Sigma_k).$$

Thus, it comes down to finding β_{ik}, μ_k, and Σ_k. The solutions or estimates must be obtained for every i at every time step t, and they must work well in the high dimensional situation with low computational cost.

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods (with discussion). *Journal of Royal Statistical Society,* Series B, 72(3):269–342.

Bengtsson, T., Bickel, P., and Li, B. (2008). Curse-of-dimensionality revisited: Collapse of importance sampling in very large scale systems. *Probability and Statistics: Essays in Honor of David A. Freedman,* IMS Collections, 2:316–334.

Bengtsson, T., Snyder, C., and Nychka, D. (2003). Toward a nonlinear ensemble filter for highdimensional systems. *Journal of Geophysical Research*, 108:8775–8785.

Bertino, L., Evensen, G., and Wackernagel, H. (2003). Sequential data assimilation techniques in oceanography. *International Statistical Review / Revue Internationale de Statistique*, 71(2):223–241.

Bickel, P. and Levina, E. (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227.

Butala, M., Frazin, R., Chen, Y., and Kamalabadi, F. (2009). Tomographic imaging of dynamic objects with the ensemble kalman filter. *IEEE Transactions on Image Processing*, 18(7):1573–1587.

Butala, M., Yun, J., Chen, Y., Frazin, R., and Kamalabadi, F. (2008). Asymptotic convergence of the ensemble kalman filter. In *Proceedings of the 2008 IEEE International Conference on Image Processing*, 825–828.

Douc, R. (2005). Comparison of resampling schemes for particle filtering. In *In 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 64–69.

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

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Doucet, A. and Gordon, N. J. (1999). Simulation-based optimal filter for maneuvering target tracking. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, 3809:241–255.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: fifteen years later. In Crisan, D. and Rozovsky, B., editors, *Handbook of Nonlinear Filtering*, 656–704. Cambridge University Press, Cambridge.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge, UK: Cambridge University Press.

Elliott, R. J., Aggoun, L., and Moore, J. B. (1995). *Hidden Markov Models: Estimation and Control.* New York: Springer.

Evensen, G. (1994). *Data assimilation: The Ensemble Kalman Filter.* Springer.

Furrer, R. and Bengtsson, T. (2007). Estimation of high-dimensional prior and posterior covariance matrices in kalman filter variants. *Journal of Multivariate Analysis*, 98:227–255.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57:1317–1339.

Givens, G. and Hoeting, J. (2005). *Computational Statistics.* Wiley Series in Probability and Statistics.

Gordon, N., Salmond, D., and Ewing, C. (1995). Bayesian state estimation for tracking and guidance using the bootstrap filter. *Journal of Guidance, Control, and Dynamics*, 18:1434–1443.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F: Radar and Signal Processing*, 140(2):107–113.

Ikoma, N., Ichimura, N., Higuchi, T., and Maeda, H. (2001). Maneuvering target tracking by using particle filter. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 4:2223–2228. IEEE.

Kong, A., Liu, J., and Wong, W. (1997). The properties of the cross-match estimate and split sampling. *The Annals of Statistics*, 25(6):2410–2432.

Lei, J. and Bickel, P. (2009). Ensemble filtering for high dimensional non-linear state space models. Technical Report 779, Department of Statistics, UC Berkeley.

Lin, M., Zhang, J., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421.

Liu, J. and Lawrence, C. (1999). Bayesian inference on biopolymer models. *Bioinformatics*, 15:38–52.

Liu, J. S. (2001). *Monte Carlo Strategies for Scientific Computing.* New York: Springer.

Lorenz, E. (2006). Predictability: A problem partly solved. In *Predictability of Weather and Climate*, 40–58. Cambridge Univ. Press, Cambridge, U. K. Originally presented in a 1996 ECMWF workshop.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Tsay, R. S. (2002). *Analysis of Financial Time Series.* New York: Wiley.