A COMPARATIVE STUDY OF ADAPTIVE MCMC BASED PARTICLE FILTERING METHODS

By

JAE MYUNG YOON

A THESIS PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE UNIVERSITY OF FLORIDA 2012
To my wife, Jungeun Park, my son, Andrew J. Yoon, my parents, Hyungsub Yoon and Heeja Yang, my in-laws, Wonrak Park and Younghee Bae, and the rest of my family members for their unwavering support and constant encouragement.
ACKNOWLEDGMENTS

I would like to sincerely thank my advisor, Mrinal Kumar, whose infinite motivation and inspiration carried me to the successful completion of my Master of Science degree. His constant guidance and the patience over the past years helped me mature academically, professionally, and personally. I would also like to express my sense of gratitude to all my committee members Norman Fitz-Coy, and Prabir Barooah for the time and help they provided. I would like to thank my wife for her love and patience. Also, I would like to extend immeasurable gratitude to my family for supporting and believing in me.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>9</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1  INTRODUCTION</td>
<td>10</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>10</td>
</tr>
<tr>
<td>1.2 Background</td>
<td>11</td>
</tr>
<tr>
<td>2  NONLINEAR BAYESIAN TRACKING</td>
<td>16</td>
</tr>
<tr>
<td>2.1 Problem Formulation for State Estimation</td>
<td>16</td>
</tr>
<tr>
<td>2.1.1 Prediction Step</td>
<td>17</td>
</tr>
<tr>
<td>2.1.2 Update Step</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Research Issues</td>
<td>17</td>
</tr>
<tr>
<td>2.2.1 Nonlinearity</td>
<td>17</td>
</tr>
<tr>
<td>2.2.2 Dimensionality</td>
<td>18</td>
</tr>
<tr>
<td>2.2.3 Real-time</td>
<td>18</td>
</tr>
<tr>
<td>3  PARTICLE FILTERING METHODS</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Importance Sampling Monte Carlo Integration</td>
<td>19</td>
</tr>
<tr>
<td>3.2 Sequential Importance Sampling (SIS)</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1 Sequential Importance Sampling</td>
<td>21</td>
</tr>
<tr>
<td>3.2.2 Problem of SIS: Sample Degeneracy</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Generic Particle Filter (PF)</td>
<td>26</td>
</tr>
<tr>
<td>3.3.1 Resampling</td>
<td>26</td>
</tr>
<tr>
<td>3.3.2 Sequential Importance Resampling (SIR)</td>
<td>27</td>
</tr>
<tr>
<td>3.3.3 Adaptive SIR</td>
<td>28</td>
</tr>
<tr>
<td>3.4 Markov Chain Monte Carlo (MCMC) Move</td>
<td>30</td>
</tr>
<tr>
<td>3.4.1 Derivation of Metropolis-Hastings (MH) Algorithm Based MCMC Move</td>
<td>31</td>
</tr>
<tr>
<td>3.4.2 MCMC Move Particle Filter</td>
<td>32</td>
</tr>
<tr>
<td>3.5 Adaptive MCMC Move</td>
<td>33</td>
</tr>
<tr>
<td>4  RESULTS</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Synthetic Experiment</td>
<td>39</td>
</tr>
<tr>
<td>4.1.1 Model</td>
<td>39</td>
</tr>
<tr>
<td>4.1.2 Comparative Results</td>
<td>39</td>
</tr>
<tr>
<td>4.2 Bearing Only Tracking</td>
<td>41</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>4-1</td>
<td>Average of performances from each filter kinds over 50 runs</td>
</tr>
<tr>
<td>4-2</td>
<td>Average of performances from each filter kinds over 100 runs</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Graphical representation of the sequential importance sampling (SIS) particle filter (PF) (1 cycle)</td>
<td>25</td>
</tr>
<tr>
<td>3-2</td>
<td>SIS PF algorithm</td>
<td>25</td>
</tr>
<tr>
<td>3-3</td>
<td>Graphical representation of the sample degeneracy problem</td>
<td>26</td>
</tr>
<tr>
<td>3-4</td>
<td>Graphical representation of the multinomial resampling process</td>
<td>27</td>
</tr>
<tr>
<td>3-5</td>
<td>Systematic resampling algorithm</td>
<td>28</td>
</tr>
<tr>
<td>3-6</td>
<td>Graphical representation of the sequential importance resampling (SIR) PF (1 cycle)</td>
<td>29</td>
</tr>
<tr>
<td>3-7</td>
<td>SIR PF algorithm</td>
<td>29</td>
</tr>
<tr>
<td>3-8</td>
<td>Adaptive SIR particle filter algorithm</td>
<td>30</td>
</tr>
<tr>
<td>3-9</td>
<td>Graphical representation of the Markov chain Monte Carlo (MCMC) move PF (1 cycle)</td>
<td>33</td>
</tr>
<tr>
<td>3-10</td>
<td>MCMC move algorithm</td>
<td>34</td>
</tr>
<tr>
<td>3-11</td>
<td>MCMC move PF algorithm</td>
<td>35</td>
</tr>
<tr>
<td>3-12</td>
<td>Graphical representation of the adaptive MCMC move PF (1 cycle)</td>
<td>37</td>
</tr>
<tr>
<td>3-13</td>
<td>Adaptive MCMC move algorithm</td>
<td>38</td>
</tr>
<tr>
<td>4-1</td>
<td>Trajectories of the synthetic experiment model</td>
<td>41</td>
</tr>
<tr>
<td>4-2</td>
<td>Sample diversity comparison of the bearing only tracking model</td>
<td>44</td>
</tr>
<tr>
<td>4-3</td>
<td>Trajectories of the bearing only tracking model with various filter methods.</td>
<td>45</td>
</tr>
</tbody>
</table>
In this thesis, we present a comparative study of conventional particle filtering (PF) algorithms for tracking applications. Through the review from the generic PF to more recent Markov Chain Monte Carlo (MCMC) based PFs, we will revisit the sample impoverishment problem. For all PF methods using resampling process, maintaining appropriate sample diversity is a big problem. Although Gilks et al. proposed a MCMC based PF to avoid this problem, their method sometimes fails due to small process noise. Therefore, we propose an improved MCMC move PF method which employs an adaptive MCMC move. This adaptive MCMC process elastically manages the MCMC proposal density function to circumvent the sample impoverishment problem efficiently and gives better sample diversity for posterior approximation.
CHAPTER 1
INTRODUCTION

1.1 Motivation

Recently, various estimation strategies for nonlinear systems have been utilized in various fields of science, finance and engineering. The application domain of estimation technique is highly varied and sometimes also closely connected with our daily lives, for example:

1. From security devices such as surveillance cameras to the “artificial intelligence focus” (AI focus) features in the latest handheld digital cameras, target tracking is heavily used. The AI focus feature automatically adjusts the focal length to maintain focus on a possibly moving target.

2. Vision systems and autonomous navigation systems found in ground vehicles, aircraft, spacecraft and robots use various target tracking algorithms and are currently widely researched.

3. Tracking algorithms also find wide use in the stock market, futures trading market, as well as forecasting of various economic indicators.

4. Tracking/filtering also finds widespread use in weather forecasting; as well as prediction of life expectancy of manufacturing tools/products.

Despite its widespread applicability as enumerated above, it is extremely difficult in reality to obtain accurate estimates of the state of nonlinear systems. We define optimal state estimation (also known as filtering) as the problem of finding the “best estimate” of the states of an evolving dynamic system. Two main sources provide state information:

1. Physical laws believed to govern dynamical behavior of the system and;

2. Sensor measurements that provide direct information about system behavior.

It is important to mention that both the above sources of information are imperfect and may be corrupted by varying degrees of noise (known as process noise and measurement noise respectively). Based on the required level of accuracy, an appropriate tracking algorithm can be constructed that performs a fusion of information obtained from the above two sources. In the above structure of state tracking, numerous poorly known parameters appear as well. However, we will not be concerned with the problem of estimating such parameters in this thesis.
Physical laws governing system behavior are typically modeled as stochastic differential equations (SDEs). These equations comprise of a deterministic part and a random part (typically called process noise). Depending on the level of accuracy required or computing capability available, the deterministic part can vary in complexity. If simplified models are used for the deterministic portion, then the process noise can be appropriately adjusted (increased) to account for loss of information. In addition to physical laws, tracking algorithms utilize sensor information. The measurement models can also vary in complexity and is invariably corrupted by sensor noise.

Based on the dynamical and measurement models used, a wide variety of tracking algorithms can be utilized. It must however be noted that each algorithm is based on certain assumptions about the nature of underlying models. If any of the assumptions break down, the tracking algorithm cannot guarantee the desired accuracy. For example, the Kalman filter is the optimal tracking algorithm only if in addition to other assumptions, both the dynamical model and measurement model are linear. Despite of the fact that the Kalman filter is the most widely used family of tracking algorithms, its underlying assumptions can make it vulnerable. In this thesis, a different family of tracking algorithms called particle filters are explored. The idea is to increase accuracy while minimizing assumptions on the dynamical and measurement models.

1.2 Background

Ever since its inception, Kalman filter (KF)[16] has been the most widely used estimation strategy due to its elegant recursive form, computational efficiency and ease of implementation. Therefore, even though KF only tracks the first two moments of the state (mean and covariance), it is extremely powerful. Furthermore, KF is optimal if the following assumptions hold:

1. The dynamic system model is a linear function of the state and process noise.
2. The measurement model is also linear function of the state and measurement noise.
3. The process noise and the measurement noise are mutually independent and zero-mean Wiener process with known covariance.
4. The posterior density is Gaussian.
Unfortunately, most real life situations do not uphold all of the above assumptions. As a result, a nonlinear version of KF was required. The local linearized version of KF, known as the Extended Kalman Filter (EKF)\cite{13} was proposed. In the EKF, the nonlinear system is replaced with a first order perturbation model obtained via Taylor Series expansion about a reference trajectory. While EKF extends the applicability of the Kalman filter to nonlinear dynamic systems and measurement models, it performs poorly when the higher order terms start to dominate (e.g. when perturbation grows in magnitude over time due to nonlinear effects). To overcome this weakness, higher-order EKF\cite{22} algorithms using the second or even higher order terms in the Taylor series expansion were introduced. However, the fundamental problem remains the same, i.e. divergence due to nonlinear effects can only be delayed but not avoided.

Overall, the class of EKF algorithms have the following general shortcomings.

1. Linearization or high order system approximations are prone to divergence dependent on degree of system nonlinearity.

2. Linearized transformation sometimes cannot be obtained through Jacobian matrix, e.g. if the system contains a discontinuity, the Jacobian does not exist.

3. Computing the Jacobian matrix can be tedious, often involving excessive and complex algebraic manipulations. Also, the computation process is prone to human error.

In the higher-order EKF algorithms, these problems can become more serious.

Recently, Julier and Uhlmann proposed another filtering strategy based on the so called unscented transformation (UT) sampling and the Kalman filter framework. This filter is known as the unscented Kalman filter (UKF)\cite{15}. As opposed to EKF, which performs local linearization, the UKF is based on the principle of statistical linearization. By virtue of statistical linearization, the first two moments of the system derived from the sample points match exactly with the first two moments of the actual system. To perform the unscented transformation, a minimal set of sample points (called sigma points) are deterministically selected around the currently known mean of the state. The sigma points are then propagated through the nonlinear system dynamics (i.e. without analytic local linearization). As a consequence, the UKF algorithm is applicable to
systems with discontinuities. However, if nonlinear effects are strong, the UKF strategy may not be adequate to describe state uncertainty.

Another popular filtering strategy is the sequential Monte Carlo (SMC) method (also known as particle filter (PF))[1, 4, 7, 26]. The basic idea behind the SMC is to characterize state uncertainty in terms of a finite number of particles. As opposed to the UKF, the SMC methods can capture higher order statistical information of the system state (e.g. the mean, variance, skewness, kurtosis etc.), by analysis of the particle statistics.

Even though the first versions of the SMC methods can be found in papers dating back to the 1950s[10], they could not become popular immediately due to the following drawbacks:

1. The SMC methods generally requires high computing power, which have only recently became readily available.
2. The early SMC methods were based on the sequential importance sampling (SIS) technique, which suffers from a serious problem called “sample degeneracy (a.k.a. particle depletion)”.

In 1993, Gordon, Salmond and Smith proposed the sequential importance resampling (SIR)[9] algorithm to overcome sample degeneracy. Since introduction of the SIR filter, research in the SMC methods has grown vigorously and resulted significant theoretical progress. In addition, thanks to the recent computer revolution, the SMC methods became increasingly amenable to the online demands of the filtering problem. After the SIR filter was introduced many other resampling ideas were proposed and implemented to improve the SIR filter. For example, multinomial resampling[9], residual resampling[20], stratified resampling[17], and systematic resampling[17] algorithms have been proposed by many authors. Following several years of implementation, the general consensus appears to be that owing to its simple implementation, low computation complexity and low sample variance, the systematic resampling algorithm is the most widely used resampling technique.

Despite the fact that resampling alleviates the problem of sample degeneracy, another serious phenomenon came up to the surface, called as “sample impoverishment”. In this phenomenon, particles with low weights disappear during the resampling step, while particles
with higher weights replicate. In an extreme case, a set of $N$ distinct particles can be reduced to a single particle repeated $N$ times. Since loss of sample diversity represents loss of statistical information of the system state, it may cause very harmful effects on state estimation. Therefore, appropriate step must be taken to maintain sample diversity. Recently, two major techniques were proposed in literature to improve the particle diversity:

1. **Regularized particle filter (RPF)**\[25\]: RPF resamples form a continuous approximation of the posterior, as opposed to the SIR PF which does so from a discrete approximation. The continuous approximation of posterior density is constructed by the regularization technique, which builds a Gaussian kernel based on the existing particles.

2. **MCMC move technique**: Proposed by Gilks et al. in 2001\[8\] is another potential solution to the sample impoverishment problem. The MCMC move algorithm uses Markov chain Monte-Carlo (MCMC) method\[27\], to relocate (“move”) existing particles to better represent (i.e. with greater diversity) the posterior density. Typically the Metropolis-Hastings (MH) algorithm\[12, 24\] is used for this purpose.

Above two different approaches are mainly used to improve sample diversity. However, we will not be concerned with the RPF in this thesis because our research focus is an extension of the MCMC move. Even though the MCMC move strategy was proposed to give sample diversity, it can break down if process noise is very small\[14\].

In this thesis, we propose a new MCMC move algorithm, named adaptive MCMC move particle filter to address the low process noise scenario. As mentioned above, when the process noise is very small, the MCMC move PF sometimes fails to track a moving object\[14\] due to the sample impoverishment problem. In the Metropolis-Hastings MCMC algorithm\[12, 24\], samples states are first proposed from a proposal density function, and then accepted or rejected. In the related field of MCMC based integration, it has been seen that adaptively adjusting the support of the proposal density function improves the performance of MCMC integration\[28\].

In this adaptive approach, the support adjustment is performed such that the acceptance rate is maintained, between 0.4 and 0.5. The current work draws inspiration from this idea and adapts it to the PF framework. In the proposed method, the ratio of number of particles actually moved by the standard MCMC move method to the total number of particles is used as an indicator of the quality of sample diversity as in MCMC integration; if this ratio lies outside a desired window
(depending on the problem), the covariance matrix of the process noise is adjusted to propose a new set of moved particles. Note that the actual system has not been altered. The process noise covariance is manipulated only in order to propose a new set of moved points. These may be accepted or rejected based on their likelihood of belonging to the actual posterior pdf.

The remainder of this thesis is organized as follows. In chapter 2, the problem formulation for the Bayesian state estimation is presented. In chapter 3, the particle filter (PF) and its variants are briefly reviewed, including the associated algorithms for practical implementation. The above mentioned new adaptive MCMC move algorithm is developed and its detailed algorithm is provided. In chapter 4, we discuss examples for practical implementation. Lastly, chapter 5 contains conclusion and paths for future work.
CHAPTER 2  
NONLINEAR BAYESIAN TRACKING

2.1 Problem Formulation for State Estimation

Consider the following discrete-time, hidden Markov model which is called the state transition model and the state measurement model:

\[
X_k | (X_{k-1} = x_{k-1}) \triangleq p(x_k | x_{k-1}) \quad (2-1)
\]

\[
Z_k | (X_k = x_k) \triangleq p(z_k | x_k) \quad (2-2)
\]

where \( X_k = \{x_0, x_1, \ldots, x_k\} \) is the sequence of system states up to time \( k \in \mathbb{N} \), and \( Z_k = \{z_1, z_2, \ldots, z_n\} \) is the sequence of measurement available up to time \( k \). Also, \( x_k \in \mathbb{R}^{n_x} \) denotes the state of the system at time \( k \), \( z_k \in \mathbb{R}^{n_z} \) denotes the observation at time step \( k \) and \( n_x, n_z \) denote the dimensions of the state vector and measurement respectively. We assume that \( X_k \) and \( Z_k \) are statistically independent. The state transition and the state measurement models (eqn. (2–1) and eqn. (2–2)) can be rewritten in functional form as shown below:

\[
x_k = f_{k-1}(x_{k-1}, v_{k-1}) \quad (2–3)
\]

\[
z_k = h_k(x_k, w_k) \quad (2–4)
\]

where \( v_k \in \mathbb{R}^{n_v} \) denotes the process noise at time \( k \), \( w_k \in \mathbb{R}^{n_w} \) denotes the measurement noise at time \( k \) and \( n_v, n_w \) denote the dimensions of the process noise and measurement noise respectively. Note that the initial state distribution, \( p(x_0) \triangleq p(x_0 | z_0) \), is assumed known.

The objective of the filtering problem is to obtain the estimated state, \( x_k \), given all available information up to time \( k \); the measurement information \( Z_k \) grows sequentially like a filtration, meaning that \( Z_k \) is not known at time step \( k - 1 \). Then, the marginal pdf of the state \( p(x_k | Z_k) \) can be computed recursively in two stages; namely, prediction and update.
2.1.1 Prediction Step

Suppose the state estimate at the previous time step \((k - 1)\) is known; i.e. \(p(x_{k-1}|Z_{k-1})\).

Then the prior pdf of the state is obtained as:

\[
p(x_k|Z_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z_{k-1})dx_{k-1}
\] (2–5)

The above equation is known as the Chapman-Kolmogorov equation (CKE), which in differential form is nothing but the Fokker-Plank equation (FPE). Note that the CKE is more general than FPE because it holds for both continuous and discrete systems.

2.1.2 Update Step

At the current time \(k\), the posterior pdf can be obtained by incorporating the new measurement via the Bayes rule as follow:

\[
p(x_k|Z_k) = \frac{p(z_k|x_k, Z_{k-1})p(x_k|Z_{k-1})}{p(z_k|Z_{k-1})} = \frac{p(z_k|x_k)p(x_k|Z_{k-1})}{p(z_k|Z_{k-1})}
\] (2–6)

where the normalizing constant can be obtained as:

\[
p(z_k|Z_{k-1}) = \int p(z_k|x_k)p(x_k|Z_{k-1})dx_k
\] (2–7)

2.2 Research Issues

2.2.1 Nonlinearity

If \(f(\cdot, \cdot)\) and \(h(\cdot, \cdot)\) are linear and \(v_k\) and \(w_k\) are Gaussian then the recursive state estimation can be solved analytically. If not (this is the typical real-life situation), filtering becomes an infinite dimension problem (equivalent to estimating all moments of the state pdf). This is an intractable problem and sub-optimal methods must be developed[19].
2.2.2 Dimensionality

In any filtering application, the “curse of dimensionality” refers to various negative phenomena occurring due to analysis of high-dimensional spaces. One of the common manifestations of this problem is that, when \( n_x \) (i.e. state dimensionality) increases, the numerical implementation of both prediction (CKE) and update (Bayes’ rule) becomes increasingly (exponentially) difficult[19].

2.2.3 Real-time

State estimation is especially difficult because it must be performed in “real time”; i.e. the prediction (CKE) and update (Bayes’ rule) stages must be completed before the next set of measurements become available. This places special efficiency demands on the developed algorithms[19].
CHAPTER 3
PARTICLE FILTERING METHODS

The sequential Monte Carlo (SMC) methods are a class of sub-optimal filters. They are
known variously as the bootstrap filter[9], the condensation algorithm[21], the particle filter
(PF)[3] etc. In order to describe SMC, we will begin with a discussion of the “importance
sampling” based Monte Carlo integration technique. Then, we will review the details of the
conventional PF methods, namely the sequential importance sampling (SIS, one of the first
versions of PF) and the sequential importance resampling (SIR) particle filter. The Markov chain
Monte Carlo (MCMC) based PF will then be introduced to overcome the shortcomings of the
SIR PF. Finally, we will describe the new adaptive MCMC move method to obtain improved
sample diversity.

3.1 Importance Sampling Monte Carlo Integration

Suppose we want to integrate a function \( g(x) \),

\[
I = \int g(x) \, dx
\]  

where \( x \in \mathbb{R}^n \). We consider the special case in which the integrand \( g(x) \) can be factored as
follows:

\[
g(x) = f(x) \cdot \pi(x)
\]  

where \( \pi(x) \) is a probability density function (pdf) satisfying the properties, \( \pi(x) \geq 0 \) and
\( \int \pi(x) \, dx = 1 \). If \( \pi(x) \) is readily available and easy to sample from; i.e. we can find a set of
points (“particles), \( \{x^i, i = 1, \ldots, N\} \sim \pi(x) \), \( N \gg 1 \); we can decompose the function using \( I \) as
follows:

\[
I_N = \frac{1}{N} \sum_{i=1}^{N} f(x^i)
\]  

where, \( N \) represents the number of samples. This is the well known Makov chain Monte Carlo
(MCMC) method of numerical integration. Moreover, if the samples, \( x^i \sim \pi(x) \), for \( i = 1, \ldots, N \)
are independent and identically distributed (iid) random variables, then $E[I_N - I] = 0$, as $N \to \infty$. $I_N$ will converge almost surely to $I$ according to the law of large numbers. The variance of $f(x)$ can be given as:

$$\sigma^2 = \int (f(x) - I)^2 \cdot \pi(x) dx.$$  

(3–4)

If the variance is finite, the estimation error converges to normal distribution with mean 0 and $\sigma^2$,

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

(3–5)

Unfortunately, $\pi(x)$ is invariably not known, or not easy to sample from. In this situation, an “importance density”, $q(x)$ is introduced, and is also known as the proposal density. We can rewrite eqn.(3–1) as follows:

$$I = \int f(x) \cdot \pi(x) dx = \int f(x) \cdot \frac{\pi(x)}{q(x)} \cdot q(x) dx.$$  

(3–6)

The proposal $q(x)$ is specifically chosen to be easy to sample from and must satisfy some basic requirements[4], e.g.:

$$q(x) \geq c \cdot \pi(x).$$  

(3–7)

Now we can estimate $I$ using samples drawn from $q(x)$, i.e. $\{x^i, i = 1, ..., N\} \sim q(x)$ as follows:

$$I_N = \frac{1}{N} \sum_{i=1}^{N} f(x^i) \cdot \tilde{w}(x^i)$$  

(3–8)

where $\tilde{w}(x^i)$ are the unnormalized importance weights and given by:

$$\tilde{w}(x^i) = \frac{\pi(x^i)}{q(x^i)}, \quad \text{for } i = 1, ..., N.$$  

(3–9)
Note that these discrete weights may not automatically be normalized: i.e. \( \sum_{i=1}^{N} \tilde{w}(x^i) \neq 1 \). To maintain the unbiased nature of the approximation, the weights must be normalized, resulting in the final approximation as follows:

\[
I_N = \frac{1}{N} \sum_{i=1}^{N} f(x^i) \cdot \tilde{w}(x^i) = \sum_{i=1}^{N} f(x^i) \cdot \frac{\tilde{w}(x^i)}{\sum_{j=1}^{N} \tilde{w}(x^j)}
\]

(3–10)

where, the normalized weight can be computed as:

\[
w(x^i) = \tilde{w}(x^i) \sum_{j=1}^{N} \tilde{w}(x^j).
\]

(3–11)

If the importance density (proposal density) is not chosen properly\[^4\], the above approach can result in particles with very low (negligibly low) weights. This affects the quality of approximation and will manifest in the form of “sample degeneracy” in the importance sampling based PF described next.

3.2 Sequential Importance Sampling (SIS)

3.2.1 Sequential Importance Sampling

Essentially, SIS is a sequence of the basic importance sampling Monte Carlo integration step described above. The unknown pdf \( \pi(x) \) is the underlying state pdf which must be approximated using particles. Since it cannot be directly sampled from; a sequence of importance density is used to generate particles with above weights. Therefore, let \( \{X_k^i, w_k^i\}_{i=1}^{N} \) denote a random measure that characterizes \( p(X_k|Z_k) \), where \( p(X_k|Z_k) \) represents the joint posterior density at time \( k \); \( \{X_k^i\}_{i=1}^{N} \) is the set of support points with associated weights \( \{w_k^i\}_{i=1}^{N} \) which satisfy \( \sum_{i=1}^{N} w_k^i = 1 \) (the normalized weights). The joint posterior density at time \( k \) can now be depicted as follows\[^{26}\]:

\[\]
\[ p(X_k|Z_k) \approx \sum_{i=1}^{N} w^i_k \delta(X_k - X^i_k) \] (3–12)

where \( \delta(\cdot) \) is the Dirac delta function and \( \delta(X_k - X^i_k) \) denotes the Dirac delta mass located at support \( X^i_k \). The above is a discrete weighted. The normalized weights, \( w^i_k \) are selected according to eqn.(3–9) as :

\[ w^i_k \propto \frac{p(X^i_k|Z_k)}{q(X^i_k|Z_k)} \] (3–13)

Since filtering is a recursive state estimation technique, assume that we have samples describing the previous joint posterior, \( p(X_{k-1}|Z_{k-1}) \). Now, we wish to approximate the joint posterior \( p(X_k|Z_k) \) by incorporating the new measurements obtained at time \( k \). If the importance density is defined as:

\[ q(X_k|Z_k) \triangleq q(x_k|X_{k-1}, Z_k)q(X_{k-1}|Z_{k-1}), \] (3–14)

then we can obtain importance samples at current time \( k \) (i.e. \( X^i_k \sim q(X_k|Z_k) \)) by combining existing samples such as \( X^i_{k-1} \sim q(X_{k-1}|Z_{k-1}) \) and \( x^i_k \sim q(x_k|X_{k-1}, Z_k) \). Before we obtain the recursive form of the importance weight (weight updating), the recursive form of the joint posterior \( p(X_k|Z_k) \) must be obtained as below:

\[
\begin{align*}
p(X_k|Z_k) &= \frac{p(X_k, Z_k)}{p(Z_k)} \\
&= \frac{p(X_k, z_k, Z_{k-1}) p(X_k, Z_{k-1})}{p(X_k, Z_{k-1}) p(Z_{k-1})} \\
&= \frac{p(Z_k)}{p(Z_{k-1})} \\
&= \frac{p(z_k|X_k, Z_{k-1}) p(X_k|Z_{k-1})}{p(z_k|Z_{k-1})} \\
&= \frac{p(z_k|X_k, Z_{k-1}) p(x_k|X_{k-1}, Z_{k-1}) p(X_{k-1}|Z_{k-1})}{p(z_k|Z_{k-1})}.
\end{align*}
\] (3–15)

In the eqn.(3–15), we can regard:
\begin{align}
p(x_k|X_{k-1}, Z_{k-1}) &= p(x_k|x_{k-1}) 
\text{(3–16)}
p(z_k|X_k, Z_{k-1}) &= p(z_k|x_k), \quad \text{(3–17)}
\end{align}

since \(p(x_k|X_{k-1}, Z_{k-1})\) and \(p(z_k|X_k, Z_{k-1})\) are Markov process. Plugging the eqn. (3–16) and eqn. (3–17) into the eqn. (3–15) yields the recursive joint posterior, \(p(X_k|Z_k)\) as follows:

\begin{align}
p(X_k|Z_k) &= \frac{p(z_k|x_k)p(x_k|x_{k-1})}{p(z_k|Z_{k-1})}p(X_{k-1}|Z_{k-1}) \quad \text{(3–18)}
&\propto p(z_k|x_k)p(x_k|x_{k-1})p(X_{k-1}|Z_{k-1}). \quad \text{(3–19)}
\end{align}

Substituting the importance density function (eqn. (3–14)) and the recursive joint posterior (eqn. (3–19)) into the eqn. (3–13) yields the desired weight updating equation as:

\begin{align}
w_k^i &\propto \frac{p(X_k^i|Z_k)}{q(X_k^i|Z_k)} 
&= \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|X_{k-1}^i, Z_k)} \cdot \frac{p(X_{k-1}^i|Z_{k-1})}{q(X_{k-1}^i|Z_{k-1})} 
&= w_{k-1}^i \cdot \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|X_{k-1}^i, Z_k)}. \quad \text{(3–20)}
\end{align}

Moreover, in the eqn. (3–20), if \(q(x_k|X_{k-1}, Z_k) = q(x_k|x_{k-1}, z_k)\), then the importance weight is only relative to \(x_{k-1}\) and \(z_k\) and can be represented as as follows:

\begin{align}
w_k^i &\propto w_{k-1}^i \cdot \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \quad \text{(3–21)}
\end{align}

From the eqn. (3–12), the discrete approximation of the marginal posterior density, \(p(x_k|Z_k)\), can then be obtained as below:
\[ p(x_k|Z_k) \approx \sum_{i=1}^{N} w_i^k \delta(x_k - x_i^k). \]  

(3–22)

SIS particle filtering algorithm recursively propagate \( N \) support points and associated importance weights (i.e. \( \{x_i^k, w_i^k\}_{i=1}^N \)) using eqn.(3–19) and eqn.(3–20). The above SIS PF process lays the groundwork of most PF filters.

Note that a selection of the importance density function is a critical design issue of any PF system. The optimal importance density function is only available in limited cases[5, 6]. Therefore, we will use the most commonly used suboptimal importance density, the transitional prior, as follows:

\[ q(x_k|x_{k-1}^i, z_k) \overset{\Delta}{=} p(x_k|x_{k-1}^i). \]  

(3–23)

Plugging the transional prior into the weight updating eqn.(3–20) yields:

\[
w_k^i \propto w_{k-1}^i \cdot \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} = w_{k-1}^i \cdot p(z_k|x_k^i). 
\]  

(3–24)

See the fig.(3-1) describing one cycle of above SIS PF process graphically. Also, see the details of the SIS PF algorithm for practical implementation is presented in the fig.(3-2)[26]

3.2.2 Problem of SIS: Sample Degeneracy

Even though the ideal importance density function is the posterior density function itself, we are not normally able to draw importance weights directly from the posterior. If the importance density \( q(x) \) in not chosen properly (i.e. eqn.(3–7)), the variance of the importance weights only increases over the time; for example, see Ref[5]. The variance increment leads to a potentially risky phenomenon of the SIS PF, called the sample degeneracy problem. After a certain number of recursive cycles, only one particle occupies the most of the weight and the rest of the particles
Figure 3-1. Graphical representation of the sequential importance sampling (SIS) particle filter (PF) (1 cycle)

Algorithm

1. FOR i=1:N
   (a) Draw \( x^i_k = q(x^i_k | x^i_{k-1}, z_k) \)
   (b) Evaluate the unnormalized importance weights via eqn.(3–20) or eqn.(3–24)
      \[ w^i_k \propto w^i_{k-1} \cdot \frac{p(z_k | x^i_k)p(x^i_k | x^i_{k-1})}{q(x^i_k | x^i_{k-1}, z_k)} \]

2. END FOR

3. Normalize the importance weight: \( \{ w^i_k \}_{i=1}^N = \frac{\{ \tilde{w}^i_k \}_{i=1}^N}{\sum_{j=1}^N \tilde{w}^j_k} \)

Figure 3-2. SIS PF algorithm

become negligible (see the fig.(3-3)). In the SIS PF framework, the sample degeneracy problem is inevitable. Therefore, a resampling technique was introduced to handle this problem.
3.3 Generic Particle Filter (PF)

3.3.1 Resampling

Basically, the resampling process involves mapping the random measures \( \{x^{(i)}_k, w^{(i)}_k\}_{i=1}^N \) into a new uniformly weighted random measures \( \{x^{(j)}_k, \frac{1}{N}\}_{j=1}^N \). In other words, the resampling process uniformly draws a sample \( x^{(i)}_k \) with probability of \( w^{(i)}_k \) from the discrete set \( \{x^{(i)}_k, w^{(i)}_k\}_{i=1}^N \). After resampling process, particles with low weights disappear while particles with higher weights replicate. Moreover, the weight set \( \{w^{(i)}_k\}_{i=1}^N \) is normalized as \( \{w^{(j)}_k = \frac{1}{N}\}_{i=1}^N \).

Figure (3-4)[23] depicts the multinomial resampling method which is presented in the paper of Gordon, Salmond and Smith[9]. In the multinomial resampling process, the cumulative distribution function (cdf) is constructed with the importance weights, \( \{w^{(i)}_k\}_{i=1}^N \). Initially, uniformly drawn \( i^{th} \) sample (i.e. \( u^i \sim U(0, 1) \), for \( i = 1, ..., N \)) is projected onto the distribution range. The intersection of \( u^i \) is then projected again onto the distribution domain to get the new sample index \( j \). This depicts how the new particle set \( \{x^{(j)}_k\}_{j=1}^N \) and the index of parents set \( \{i^j\}_{j=1}^N \) are obtained. After the resampling step was first implemented to the SIS filter, many other resampling ideas were proposed to improve SMC methods. For example, multinomial...
resampling[9], residual resampling[20], stratified resampling[17], and systematic resampling[17] algorithms have been proposed by many authors. Following several years of implementation, the general consensus appears to be that owing to its simple implementation, low computation complexity and low sample variance, the systematic resampling algorithm is the most widely used resampling technique. In this thesis, we will only review the systematic resampling algorithm which is presented below in fig.(3-5)[26]. Generally in any particle filter systems, the index of the sample parent, $i^j$, is not required except for auxiliary PF and the MCMC move based PF.

### 3.3.2 Sequential Importance Resampling (SIR)

We now present one cycle of the generic PF process (SIR PF); See the fig.(3-6). In this process, we considered the transition prior is considered as the importance density (i.e. $q(x_k|x_{k-1}^i, z_k) \triangleq p(x_k|x_{k-1}^i)$). At the top of the figure, we start with the posterior approximation at time $k - 1$, i.e. $\{x_{k-1}^i, \frac{1}{N}\}_{i=1}^N$, which is uniformly weighted. In the prediction step, above particles are propagated through the system dynamics (eqn.(2–3)) and then, the particles
Algorithm

\[ \{x_k^j, w_k^j, i^j\}_{j=1}^N = \text{RESAMPLE}\{x_k^i, w_k^i\}_{i=1}^N \]

1. Construct the cdf \( \{e^i\}_{i=1}^N \) with particles’ importance weight \( \{w_k^i\}_{i=1}^N \)
2. Set the cdf index: \( i = 1 \)
3. Draw a random sample: \( u_1 \sim U(0, \frac{1}{N}) \)
4. FOR \( j=1:N \)
   (a) Move along the cdf: \( u_j = u_i + \frac{j-1}{N} \)
   (b) WHILE \( u_j > c_i \)
       i. \( i = i + 1 \)
   (c) END WHILE
   (d) Assign sample: \( x_k^j = x_k^i \)
   (e) Assign weight: \( w_k^j = \frac{1}{N} \)
   (f) Assign parent: \( i^j = i \)
5. END FOR

Figure 3-5. Systematic resampling algorithm

approximate the prior density, \( \{x_k^i, \frac{1}{N}\}_{i=1}^N \approx p(x_k|Z_{k-1}) \). When a new observation \( z_k \) is available, we update the importance weight of each particles such that the particles approximate the marginal posterior, \( \{x_k^i, w_k^i\}_{i=1}^N \approx p(x_k|Z_k) \). Then we execute the resampling process to circumvent the sample degeneracy problem. During the resampling process, the particles with low weights are removed while particles with higher weights are duplicated. The resampled particles still approximate the marginal posterior at the current time \( k \) as \( \{x_k'^i, \frac{1}{N}\}_{i=1}^N \approx p(x_k|Z_k) \).

The details of the SIR PF algorithm for practical implementation is presented in the fig. (3-7)[26]

### 3.3.3 Adaptive SIR

The only difference between the SIR filter and the adaptive SIR filter is the existance of computing the effective sample size (\( \hat{N}_{\text{eff}} \)) step. In the standard SIR filter process, the resampling process is performed every time step (i.e. time \( =1, \ldots, k \)) after every weight updating
Algorithm

\[
\{x^i_k\}_{i=1}^N = \text{SIR}[\{x^i_{k-1}\}_{i=1}^N, z_k]
\]

1. FOR \(i = 1 : N\)
   (a) Draw \(x^i_k \sim p(x_k|x^i_{k-1})\)
   (b) Compute the unnormalized importance weight: \(\tilde{w}^i_k = p(z_k|x^i_k)\)
2. END FOR
3. Normalize the importance weights: \(\{w^i_k\}_{i=1}^N = \frac{\{\tilde{w}^i_k\}_{i=1}^N}{\sum_{j=1}^N \tilde{w}^j_k}\)
4. Resample using the resampling algorithm in fig.(3-5)
   (a) \(\{x^i_k, \cdot, \cdot\}_{i=1}^N = \text{RESAMPLE}[\{x^i_k, w^i_k\}_{i=1}^N]\)

Figure 3-7. SIR PF algorithm

step. However, in the adaptive SIR filter process, we compute the effective sample size \(\hat{N}_{\text{eff}}\) as an indicator of the sample degeneracy. If \(\hat{N}_{\text{eff}}\) falls below the threshold of particle population \(N_{\text{thr}}\) (i.e. \(\hat{N}_{\text{eff}} \leq N_{\text{thr}}\)), the adaptive SIR filter performs the resampling step so as to circumvent the sample degeneracy. In the adaptive SIR PF, \(N_{\text{thr}}\) is an users’ choice of design parameter. The effective sample size is introduced in the paper[18] and computed as follows:
Algorithm

\[ \{x_k^i, w_k^i\}_{i=1}^N = \text{ASIR}\{\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\} \]

1. Filtering via SIS (fig.(3-2))
   \[ \{x_k^i, w_k^i\}_{i=1}^N = \text{SIS}\{\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\} \]

2. Calculate \( \tilde{N}_{eff} \) using eqn.(3–25)
   \[ \tilde{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2} \]

3. IF \( \tilde{N}_{eff} < N_{thr} \)
   
   (a) Resample using the algorithm in fig.(3-5)
   \[ \{x_k^i, w_k^i, -\}_{i=1}^N = \text{RESAMPLE}\{\{x_k^i, w_k^i\}_{i=1}^N\} \]

4. END IF

Figure 3-8. Adaptive SIR particle filter algorithm

\[ \tilde{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}. \] (3–25)

The details of the adaptive SIR algorithm for practical implementation are presented in the fig.(3-8)[26]

### 3.4 Markov Chain Monte Carlo (MCMC) Move

The resampling process was introduced to circumvent the sample degeneracy problem. However, another serious problem called “sample impoverishment” emerged due to this remedy. In the resampling step, because the resampled particles are drawn from a discrete density rather than a continuous density, the particles lose their diversity. If this problem is not taken care of properly, a set of \( N \) distinct particles can be reduced to a single particle repeated \( N \) times in an extreme case. This problem is especially severe when the process noise in system dynamics is very small. As described earlier, MCMC move technique is introduced to improve the particle diversity. The Metropolis-Hasting algorithm based MCMC move algorithm was proposed by Gilks et al. in Ref.[8]. Basically, the MCMC move step chooses the better particles between that resampled particles \( x_k^i \) and proposal samples \( x_k^{*i} \) by the acceptance probability \( \alpha \). Note that using
multiple MCMC cycle means that more processing time is spent but guarantees that particles are asymptotically converged to the desired posterior. The derivation[26] of the MCMC move step is presented in the next subsection. In addition, when the process noise in the system dynamic model is completely zero, then any PF methods will perform poorly because PF methods are proper in the estimation of stochastic dynamic systems[26]. However, the RPF[25] and the MCMC move PF[8] shows better performance when the problem of sample impoverishment is severe or when the process noise in the dynamic system is small.

3.4.1 Derivation of Metropolis-Hastings (MH) Algorithm Based MCMC Move

Remember that $X_k = \{x_j, j = 0, ..., k\}$ represents the sequence of all states up to time $k$; $p(X_k|Z_k)$ represents the joint posterior density at time step $k$; $p(x_k|Z_k)$ represents the marginal density at time $k$; From the recursive relationship of the joint posterior eqn.(3–18), we have the following equation:

$$p(X_k|Z_k) = \frac{p(z_k|x_k)p(x_k|x_{k-1})}{p(z_k|Z_{k-1})}p(X_{k-1}|Z_{k-1}).$$  (3–26)

After the resampling step at time step $k$, we have the index of parents $i^j$ and the resampled particles $x^{i^j}_k$. Using the index of parents $i^j$, we can obtain the resampled parents as follows:

$$x^{i^j}_{k-1} \triangleq x^i_{k-1}.\quad (3–27)$$

The most important idea is that we compare a resampled particle $x^i_k$ and a proposal sample from a transitional prior $x^{i^j}_k \sim p(x_k|x^i_{k-1})$. Then we decide particle movement with the acceptance probability when $u \leq \alpha (u \sim U(0,1))$ as:

$$\text{We accept movement } x^i_k = \begin{cases} x^{i^j}_k \text{ with probability } \alpha \\ x^i_k \text{ otherwise} \end{cases}\quad (3–28)$$

We can obtain particle samples from the posterior density using the eqn.(3–26) as:
The details of the MCMC move algorithm and the MCMC move PF algorithm for practical approximate the posterior. Then the MCMC move step compares the resampled particles \( x_k^i \) and we obtain the particles approximating the prior density \( q_k \).

### 3.4.2 MCMC Move Particle Filter

We now present one cycle of the standard MCMC move PF process (time \( k - 1 \rightarrow k \), \( N = 10 \)): See the fig.\((3-9)\). Note that the transitional prior is used as the importance density (i.e. \( q(x_k | x_{k-1}^i, z_k) \triangleq p(x_k | x_{k-1}^i) \)). Before the prediction step, we start with the uniformly weighted particles \( \{ x_k^i, \frac{1}{N} \}_{i=1}^N \) which is the discrete approximation of the posterior at time \( k - 1 \). In the prediction step, the particles will be propagated via the system dynamics \( f(\cdot) \)\(^{(eqn.\(2-3\))}\) and we obtain the particles approximating the prior density \( \{ x_k^i, \frac{1}{N} \}_{i=1}^N \approx p(x_k | Z_{k-1}) \). After a new observation \( z_k \) is obtained, we update the importance weight of each particle such that the particles approximate the posterior density, \( \{ x_k^i, w_k^i \}_{i=1}^N \approx p(x_k | Z_k) \). During the resampling step, we obtain the resampled particles \( x_k^{ij} \) and the resampled parents \( x_{k-1}^{ij} \) via eqn.\((3-27)\).

Then the MCMC move step compares the resampled particles \( x_k^i \) and the proposal samples \( (x_k^{*i} \sim p(x_k | x_{k-1}^i)) \), to select better particles via eqn.\((3-28)\). The relocated particles still approximate the posterior \( \{ x_k^i, \frac{1}{N} \}_{i=1}^N \approx p(x_k | Z_k) \) but having an improved sample diversity. The details of the MCMC move algorithm and the MCMC move PF algorithm for practical

\[
p(X_k^*|Z_k) = \frac{p(z_k|x_k^*)p(x_k^*|x_{k-1}^i)}{p(z_k|Z_{k-1})}p(X_{k-1}|Z_{k-1}) \tag{3-29}
\]

Assume that MCMC proposal density \( a(\cdot | x_k^i) \) is symmetric. Then we can cancel the MCMC proposals with the relationship \( a(x_k^*|x_k^i) = a(x_k^i|x_k^*) \). Then the acceptance probability \( \alpha \) can be computed as follows:

\[
\alpha = \min \left( 1, \frac{p(X_k^*|Z_k)a(x_k^*|x_k^i)}{p(X_k^i|Z_k)a(x_k^i|x_k^*)} \right)
\]

\[
= \min \left( 1, \frac{p(X_k^i|Z_k)}{p(X_k^i|Z_k)} \right)
\]

\[
= \min \left( 1, \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)} \right)
\]

\[
= \min \left( 1, \frac{1}{p(x_k^i|x_{k-1}^i)} \right)
\]

\[
= \min \left( 1, \frac{1}{p(x_k^i|x_{k-1}^i)} \right)
\]

\[
= \min \left( 1, \frac{1}{p(x_k^i|x_{k-1}^i)} \right)
\]

\[
= \min \left( 1, \frac{1}{p(x_k^i|x_{k-1}^i)} \right)
\]
implementation are presented in the fig.3-10 and the fig.3-11 respectively. Note that multiple cycles of the MCMC move can be repeated (e.g. \( s = S_n(S_n > 1) \)) so that the approximation of the posterior MCMC converges (e.g. \( S_n \geq S_{burn-in} \)).

### 3.5 Adaptive MCMC Move

The sample impoverishment problem is known to be severe when the process noise of the system dynamics is small[26]. In such cases, we normally choose either the regularized PF or the MCMC move PF. Even though the MCMC move PF performs better than the SIR PF, the MCMC move PF sometimes breaks down due to the sample impoverishment problem[14]. This is because the proposal samples are not generated from broad state due to the small process noise. If the MCMC move PF can adjust the process noise covariance (or standard deviation) purely for the purpose of generating better proposal state, we can have better proposal samples via the MCMC move step. Therefore, we propose a new MCMC move step with adaptive process noise.
Algorithm

\[
\{x_i^k\}_{i=1}^N = \text{MCMC}\left[\{x_{ij}^k\}_{i=1}^N, \{x_{ij}^{k-1}\}_{ij=1}^N\right]
\]

1. FOR \(s = 1 : S_n\)
   
   (a) FOR \(i = 1 : N\)
   
   i. Sample \(u \sim U(0,1)\)
   
   ii. Sample the proposal candidate \(x_k^i\) from the transitional prior
   
   iii. IF \(u \leq \alpha = \min\left(1, \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}\right)\), accept move:
   
   \[x_k^i = x_k^i\]
   
  (iv. ELSE reject move:
   
   \[x_k^i = x_k^i\]
   
   v. END IF

   (b) END FOR

2. END FOR

3. \(X_k^i = \{X_{k-1}^i, x_k^i\}\)

---

Figure 3-10. MCMC move algorithm

covariance (or standard deviation) selection in the MCMC move step to give improved sample diversity.

In the adaptive MCMC move PF, the acceptance rate, \(AR\), is used as an indicator of the sample impoverishment problem. That is, if the filter achieves lower \(AR\) than the desired \(AR\) (i.e. \(AR_{th}\)), we assume that particles are diversified enough to represent the uncertainty. On the other hand, if the filter achieves higher \(AR\) than the desired \(AR\), the filter need to adjust the proposal noise covariance (or standard deviation) so as to obtain better particle samples. \(AR\) is defined as the proportion of the accepted number of particles over the entire particle population \(N\). \(AR\) is computed as follows:

\[
AR = \frac{N_{\text{accepted}}}{N}. \tag{3-31}
\]
Algorithm

\[ \{ x_k^i \}_{i=1}^N = \text{MCMCPF/AMCMCPF}[\{ x_{k-1}^i \}_{i=1}^N, z_k] \]

1. FOR \( i = 1 : N \)
   (a) Draw \( x_k^i \sim p(x_k|x_k^{i-1}) \)
   (b) Calculate the unnormalized importance weight: \( \tilde{w}_k^i = p(z_k|x_k^i) \)

2. END FOR

3. Normalize the importance weights: \( \{ w_k^i \}_{i=1}^N = \frac{\{ \tilde{w}_k^i \}_{i=1}^N}{\sum_{j=1}^N \tilde{w}_k^j} \)

4. Resample using the algorithm in fig.(3-5)
   (a) \( \{ x_k^j, x_k^i \}_{j=1}^N = \text{RESAMPLE}[\{ x_k^i, w_k^i \}_{i=1}^N] \)

5. FOR \( j=1:N \)
   (a) Assign parents: \( x_{k-1}^j = x_k^{ij} \)

6. END FOR

7. \( \{ x_k^i \}_{i=1}^N = \text{MCMC/AdaptiveMCMC}[\{ x_k^i \}_{i=1}^N, \{ x_{k-1}^j \}_{j=1}^N] \)

Figure 3-11. MCMC move PF algorithm

The adaptive MCMC move PF method repeats \( S_n (S_n > 1) \) times of the propagation phase until the desired AR is achieved. If we get desired \( AR \), we assume the approximation of the posterior is MCMC converged (e.g. \( S_n \geq S_{\text{burn-in}} \)). It is intuitive that the repeating the propagation phase harm the processing time but benefit the filter accuracy (uncertainty information). In some cases (see Ref[14]), it has seen that repeating the standard MCMC move[8] is very slow to converge to the desired target density. However, the adaptive MCMC move PF accelerates MCMC convergence using adaptive adjustment of the proposal noise covariance (or standard deviation) step. As a result, the adaptive MCMC move PF method can achieve a better state estimation by improving sample diversity.

The adaptive adjustment of the proposal noise covariance (or standard deviation) is performed by multiplying extension magnitude, \( \eta (\eta \geq 1) \), to the proposal noise covariance
(or standard deviation). The choice of $\eta$ is very important design issue in the design of the adaptive MCMC move PF. It is recommended to designate more than 2 levels for the acceptance rate. For example:

\begin{align*}
AR > AR_1 & \rightarrow Q_k(\text{or } \sigma_k) = \eta_1 \cdot Q(\text{or } \sigma), \\
AR_1 > AR > AR_2 & \rightarrow Q_k(\text{or } \sigma_k) = \eta_2 \cdot Q(\text{or } \sigma), \\
Otherwise(AR \leq AR_{th}) & \rightarrow Q_k(\text{or } \sigma_k) = Q(\text{or } \sigma).
\end{align*}

(3–32) 
(3–33) 
(3–34)

A graphical representation of the adaptive MCMC move PF process (time $k - 1 \rightarrow k$, $N = 10$ and one cycle adaptive MCMC move phase) is presented in the fig.(3-12). The previous steps before the adaptive MCMC move is identical to the standard MCMC move; see the fig.(3-10). After one phase of the standard MCMC move, the adaptive MCMC move PF determines (based on the $AR$) if the proposal noise covariance (or standard deviation) adjustment is required or not. If a high $AR$ (i.e. higher than the desired $AR$) is obtained, extension parameter, $\eta$, will be multiplied to the noise covariance (or standard deviation). We then re-propagate the parents of resampled particles ($x^{i}_{k-1} \triangleq x^{i}_{k-1}$) to obtain better proposal samples ($x^{*i}_{k} \sim p(x_{k}|x^{i}_{k-1})$). Note that the parents of resampled particles are obtained via the index of parents. In addition, they represent the particles at the previous time step (k-1) which have survived offspring after resampling step. Next, the adaptive MCMC move PF will compare the resampled particles ($x^{i}_{k}$) to the new proposal samples ($x^{*i}_{k}$) and choose better particle samples via MH algorithm eqn.(3–28). After the adaptive MCMC move, particles approximate the posterior

$$
\{x^{i}_{k}, \frac{1}{N}\}^{N}_{i=1} \approx p(x_{k}|Z_k)
$$

with more improved sample diversity than the standard MCMC move PF. The details of the adaptive MCMC move algorithm for practical implementation are presented in the fig.(3-13). The main PF algorithm is identical to the standard MCMC move PF (see the fig.(3-11)).
Figure 3-12. Graphical representation of the adaptive MCMC move PF (1 cycle)
Algorithm

\[
[x_i^k]_{i=1}^N = \text{AdaptiveMCMC}[\{x_i^k\}_{i=1}^N, \{x_{k-1}^i\}_{i=1}^N]
\]

1. Set the initial parameters
   (a) SET the threshold acceptance rate \(AR_{th}\)
   (b) SET the extension levels using eqn.(3–32) through eqn.(3–34)

2. FOR \(s=1:S_n\)
   (a) Compare the \(AR\) to the levels, adjust proposal noise covariance (or standard deviation)
      i. IF \(AR \geq AR_1\), THEN \(Q_k = \eta_1 \cdot Q\) (see the eqn.(3.5) for design)
      ii. ELSEIF \(AR \leq AR_{th}\), THEN break
      iii. ELSE \(Q_k = Q\)
      iv. END IF
   (b) SET \(N_{\text{accepted}} = 0\)
   (c) FOR \(i = 1 : N\)
      i. Sample \(u \sim U(0, 1)\)
      ii. Sample the proposal candidate \(x_{k}^{*i}\) from the transitional prior with the proposal noise covariance \(Q_k\)
      iii. IF \(u \leq \alpha = \min\left(1, \frac{p(z_k|x_{k}^{*i})p(x_{k}^{*i}|x_{k-1}^i)}{p(z_k|x_{k}^i)p(x_{k}^i|x_{k-1}^i)}\right)\), accept move:
         A. \(x_k^i = x_{k}^{*i}\)
         B. \(N_{\text{accepted}} = N_{\text{accepted}} + 1\)
      iv. ELSE reject move:
         A. \(x_k^i = x_k^i\)
         v. END IF
   (d) END FOR
   (e) Calculate the acceptance rate: \(AR = \frac{N_{\text{accepted}}}{N}\)

3. END FOR

4. \(X_k^i = \{X_{k-1}^i, x_k^i\}\)

Figure 3-13. Adaptive MCMC move algorithm
CHAPTER 4
RESULTS

The adaptive MCMC move PF will be compared to the standard MCMC move PF and the SIR PF, in the following two examples: a synthetic experiment and the bearing only tracking model (BOTM).

4.1 Synthetic Experiment

The synthetic experiment [23] is the first example of the adaptive MCMC PF.

4.1.1 Model

For the first experiment, the system dynamic model is given as:

$$x_{k+1} = 1 + \sin(\omega \pi k) + \alpha x_k + v_k$$  \hspace{1cm} (4–1)

where $\omega$ and $\alpha$ are model parameters and given by $\omega = 0.04$ and $\alpha = 0.5$, respectively. The process noise $v_k$ is distributed with Gamma random distribution with parameters such that $\gamma(3, 2)$. And the non-stationary observation model is given as follows:

$$z_k = \beta x_k^2 + w_k, \hspace{0.5cm} (k \leq 30)$$  \hspace{1cm} (4–2)

$$= \delta x_k - 2 + w_k, \hspace{0.5cm} (k > 30)$$  \hspace{1cm} (4–3)

where $\beta$ and $\delta$ are the observation model parameters, $\beta = 0.2$, $\delta = 0.5$. The observation noise $w_k$ is distributed with the Gaussian random distribution such that $w_k = \mathcal{N}(0, 0.00001)$. Note that new observations are available every one second.

4.1.2 Comparative Results

When a sequence of observation data $\{z_k, k = 1, ..., 60\}$ is given, the state estimation $\{\hat{x}_k, k = 1, ..., 60\}$ is performed with various filtering methods. Note that the experiment data were obtained based on the average of 50 runs. Ref[23] provides an extended comparison, although the processing time for each method has not been provided. In this experiment, the second SIR PF had equivalent amount of particles compared to the processing time of the
adaptive MCMC move PF taken. The first SIR PF with $N = 20$, the second SIR PF with $N = 500$, the standard MCMC move PF with $N = 20$ and repeated multiple propagation phase $S_n = 35$ and the adaptive MCMC move particle filter with below parameters were tested:

\[
N \quad = \quad 20,
\]
\[
S_n \quad = \quad 35,
\]
\[
(AR > 70) \quad Q_k \rightarrow \eta_1 (= 3) \cdot Q,
\]
\[
(70 \geq AR > 25) \quad Q_k \rightarrow \eta_2 (= 2) \cdot Q,
\]
\[
Otherwise(AR_{th} \leq 25) \quad Q_k \rightarrow Q.
\]

In the particle filtering methods, the systematic resampling method was used. See the table.(4-1) which summarizes the performance of each particle filter method. For the comparison purposes, mean squared error (MSE) of the state estimates, variance of the MSE, and processing time are presented.

Table 4-1. Average of performances from each filter kinds over 50 runs

<table>
<thead>
<tr>
<th>Filter type</th>
<th>Avg. MSE</th>
<th>Avg. variance</th>
<th>Avg. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF ($N = 20$)</td>
<td>0.8229</td>
<td>0.0270</td>
<td>0.3369</td>
</tr>
<tr>
<td>PF ($N = 500$)</td>
<td>0.2567</td>
<td>0.0215</td>
<td>4.6724</td>
</tr>
<tr>
<td>MCMC PF ($S_n = 35, N = 20$)</td>
<td>0.5657</td>
<td>0.0344</td>
<td>5.4291</td>
</tr>
<tr>
<td>Adaptive MCMC PF ($S_n = 35, N = 20$)</td>
<td>0.1736</td>
<td>0.0010</td>
<td>5.5802</td>
</tr>
</tbody>
</table>

See the fig.(4-1) to compare the trajectories of each filtering method. Dot-dashed line shows the trajectory of SIR PF with $N = 100$, dot-dotted line shows the trajectory of generic PF (SIR) with $N = 500$, stitched line shows the trajectory of the MCMC move PF with $N = 20$ and 35 cycles, and solid line shows the trajectory of adaptive MCMC move PF method with $N = 20$, 35 cycles. Even though the adaptive MCMC move method uses only 20 particles, it shows the lowest MSE among the participating filters.
4.2 Bearing Only Tracking

For the second example, we experimented a conventional filtering application known as the bearing only tracking[2]. During the application, we track a moving object with a couple of angle sensors. Those sensors are only allowed to measure bearings (or angles) of the moving target with respect to the sensor location. The following literature contains the performance comparison of the classic filtering methods such as the extended Kalman filters (EKFs) and the unscented Kalman filters (UKFs, see the paper by J. Hartikainen et al.[11]). The state of the moving object at time step $k$ comprise of the cartesian coordinates of the target location $x_k$ and $y_k$ and the velocity of the each $x$ and $y$ directions such as $\dot{x}_k$ and $\dot{y}_k$. Then the state can be expressed as follows:
4.2.1 Model

The system dynamic model is given as

\[
\begin{bmatrix}
 x_k \\
 y_k \\
 \dot{x}_k \\
 \dot{y}_k
\end{bmatrix} =
\begin{bmatrix}
 1 & 0 & \Delta x \\
 0 & 1 & 0 \\
 0 & 1 & \Delta y \\
 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
 x_{k-1} \\
 y_{k-1} \\
 \dot{x}_{k-1} \\
 \dot{y}_{k-1}
\end{bmatrix}
+ \begin{bmatrix}
 v_{k-1}
\end{bmatrix}
\]  

(4–5)

where \( v_{k-1} \) is the process noise, which is distributed as Gaussian with zero mean. The process noise covariance can be computed as

\[
E[v_{k-1}v_{k-1}^T] =
\begin{bmatrix}
 \frac{1}{3}\Delta k^3 & 0 & \frac{1}{2}\Delta k^2 & 0 \\
 0 & \frac{1}{3}\Delta k^3 & 0 & \frac{1}{2}\Delta k^2 \\
 \frac{1}{2}\Delta k^2 & 0 & \Delta k & 0 \\
 0 & \frac{1}{2}\Delta k^2 & 0 & \Delta k
\end{bmatrix} \sigma^2
\]

(4–6)

where \( \sigma^2 \) represents the noise spectral density which is set to \( \sigma^2 = 0.2 \). The bearings observation model of the \( i^{th} \) sensor is defined as

\[
\theta_k^i = \arctan\left(\frac{y_k - s_{yi}^i}{x_k - s_{xi}^i}\right) + w_k^i
\]

(4–7)

and \( S = \begin{bmatrix} s^1 \\ s^2 \end{bmatrix} \) is the locations of the sensors matrix, where the first sensor is located at

\[
s^1 = \begin{bmatrix} s_{x1} \\ s_{y1} \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix}
\]

and the second sensor is at \( s^2 = \begin{bmatrix} s_{x2} \\ s_{y2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \).

Each observation error of the \( i^{th} \) sensor (\( w_k^i \)) is independent and distributed as Gaussian with zero
mean and variance, \( \sigma^2 \) (i.e. \( w_k = \mathcal{N}(0, \sigma^2) \)). The noise distribution parameter is given by \( \sigma = 0.01 \) radians. Note that new observations are available every 0.1 second. The initial conditions for the filtering are given as follows:

\[
\begin{bmatrix}
    x_0 = \\
    P_0 = \\
\end{bmatrix}
\begin{bmatrix}
    0 & 0 & 1 & 0 \\
    0.1 & 0 & 0 & 0 \\
    0 & 0.1 & 0 & 0 \\
    0 & 0 & 10 & 0 \\
    0 & 0 & 0 & 10 \\
\end{bmatrix}^T
\]

(4–8)

(4–9)

The initial conditions can be read that we have some information about the moving object. That is, we are convinced of the object location, but we are uncertain about the velocity of the moving object.

### 4.2.2 Comparative Results

For the second experiment, SIR PF\((N = 100)\), SIR PF\((N = 2000)\), MCMC move PF\((N = 100, S_n = 15)\) and adaptive MCMC move PF\((N = 100)\) were tested. Because the performance of the SIR PF is largely dependent on the number of particles, a large population SIR PF \((N = 2000)\) was selected for the second SIR PF. Due to the time limitation (60 seconds), any SIR PF with greater than \( N = 2000 \) was not considered. The adaptive MCMC move parameters were set as follows. Only when \( AR_1 \geq 90 \), extension magnitude \( \eta_1 = 5 \) was multiplied to the proposal noise standard deviation. Note that the experiment data were obtained based on the average of 100 runs.

See the fig.(4-2) for detail comparison of the sample diversity between the MCMC move PF and the adaptive MCMC move PF. The left hand side shows the particle distribution of the regular MCMC move PF and the right hand side shows the particle distribution of the adaptive MCMC move PF method. The intersection of stitched lines are where the measurement was made. From the right figure, the adaptive MCMC move PF is seen to work greater based on the fact that
Figure 4-2. Sample diversity comparison of the bearing only tracking model.

particles are distributed ideally over the state. Compare the result with the left figure (the regular MCMC move PF has poorer sample diversity). See the table (4-2) which is summarizing the performance of each different particle filtering methods. For the comparison purpose, root mean squared error (RMSE) of the state estimates, and processing time are presented in each column.

Table 4-2. Average of performances from each filter kinds over 100 runs

<table>
<thead>
<tr>
<th>Filter type</th>
<th>Avg. RMSE</th>
<th>Avg. variance</th>
<th>Avg. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF ($N = 100$)</td>
<td>1.1056</td>
<td>0.1131</td>
<td>2.4203</td>
</tr>
<tr>
<td>PF ($N = 2000$)</td>
<td>0.4551</td>
<td>0.0012</td>
<td>52.4790</td>
</tr>
<tr>
<td>MCMC PF w/ multiple cycles ($S_n = 15, N = 100$)</td>
<td>0.7577</td>
<td>0.0402</td>
<td>50.5498</td>
</tr>
<tr>
<td>Adaptive MCMC PF ($S_n = 15, N = 100$)</td>
<td>0.2964</td>
<td>0.0009</td>
<td>55.5341</td>
</tr>
</tbody>
</table>

Finally, the estimated state trajectories of the participating filters are seen in fig. (4-3). Dot-dotted line shows the trajectory of generic PF with $N=100$, stitched line shows the trajectory of generic PF with $N = 2000$, dot-dashed line show the trajectory of PF with MCMC move with $N = 100$, 15 cycles, and solid line shows the trajectory of PF with Adaptive MCMC method with $N = 100$, 15 cycles. From the figure, we can confirm that the trajectory of the adaptive MCMC move PF was superior to the others in spite of the small number of particles ($N = 100$). First, the SIR filter ($N = 100$) could not track the object for some time at the beginning. Even though the second SIR PF ($N = 2000$) has smoother trajectory but still has some difficulties to track the moving object completely. Next, the MCMC move PF has seen a poorer trajectory than one of
Figure 4-3. Trajectories of the bearing only tracking model with various filter methods.

the large population SIR PF($N = 2000$). Since the actual number of particles is only 100 with small process noise, the MCMC move PF is no better than the first SIR PF in the sense of sample diversity.

Remark. All of the simulation examples are executed under the following environment, MATLAB 7.9, AMD Athlon 64 Dual Core 1.7 GHz and Windows 7 Professional OS.
CHAPTER 5
CONCLUSION

5.1 Closing Statement

The conventional particle filtering algorithms are broadly used in various applications. Even though the MCMC move PF is a popular choice to avoid the sample impoverishment problem, there still exist potential dangers of failure due to sample impoverishment. In this thesis, we proposed adaptive MCMC move algorithm to avoid the sample impoverishment problem effectively by giving improved sample diversity. This new method adjusts the proposal noise covariance (or standard deviation) adaptively by the scale of acceptance rate (\(AR\)), which plays an important role: a poor sample diversity indicator. The series of experiment (given in chapter 4) showed that this adaptive MCMC move PF (with fewer particles) improves sample diversity whilst increasing the filter accuracy. In addition, in the sense of greater confidence, the adaptive MCMC move PF showed the less error variation among all of the participated particle filtering methods. Consequently, we can use this algorithm where any harsh environment is required target tracking applications (described earlier in the chapter 1 at the beginning of the motivation section) by the accuracy and reliability.

5.2 Future Work

While promise has been shown, there exists numerous avenues for future research; e.g.

1. In the recent study of MCMC integration\[28\] method, it has been seen that adaptively adjusting the support of the proposal density function improves the performance of MCMC integration. In such cases, the support of the proposal density function is adjusted in two ways: extension and shrinkage. In this thesis, we have shown only the high-\(AR\) cases (i.e. extension case). Therefore, follow-up study of the reversed cases (i.e. the low-\(AR\) cases) and their effects on the filter accuracy will provide an extension of this research.

2. Designation of the multiple \(AR\) levels was suggested in the adaptive MCMC move PF process. In this step, new extension parameter \(\eta\) was obtained via empirical method. However, if the optimal relationship between \(\eta\) and \(AR\) exists (i.e. a functional form), it would be beneficial to maintaining the appropriate sample diversity in minimal time.
REFERENCES


BIOGRAPHICAL SKETCH

Jae Myung Yoon was born in Seoul, Republic of Korea, in 1980. He received his Bachelor of Engineering degree in control engineering from Kwangwoon University, Seoul, Republic of Korea, in 2006. He then worked as an equipment development engineer at the Memory business of Samsung Electronics Co. Ltd from 2006 through 2008. He joined the University of Florida to pursue his master’s degree in mechanical engineering in 2010. He then joined the Stochastic Systems Laboratory to research under the advisement of Dr. Mrinal Kumar. He completed Master of Science from the University of Florida in May 2012.