Nonparametric Techniques for Bayesian Filtering

Proposition de sujet de recherche

Yali Wang

Directeur :

Dr. Brahim Chaib-draa

Comité d'encadrement :

Dr. Brahim Chaib-draa
Dr. Philippe Giguère
Dr. Christian Gagné

Département d'Informatique et de Génie Logiciel
Université Laval
August 2011

Résumé

Stochastic filtering refers to an issue for estimating the latent variables from the observations using a stochastic model. Since it was developed in early 1940s, filtering has been one of the most practical tools in the science and engineering field. One of the best known filtering algorithms is a class of Gaussian Filters. However, its gaussian assumption for posterior distribution severely affects their estimation performance. A class of Particle Filter is an alternative way to approximate arbitrary posterior distribution using Sequential Monte Carlo Sampling framework. Current studies have shown that the Particle Filter scheme suffers weight degeneracy problem since sequential importance sampling. Hence I propose new methods based on nonparametric technique to overcome the drawbacks in the existing filters. More specifically, I will combine different sampling methods with adaptive mechanism to improve the accuracy. Then I will design a better nonparametric proposal distribution for solving the underlying weight degeneracy problem. I will also consider bayesian nonparametric filtering and the flexibility that it offers. In this context, I will design a general filtering and smoothing framework for bayesian nonparametric models. Finally, due to the fact that all the techniques for state estimation are the foundation of nonlinear control problems, I will incorporate the nonparametric filtering methods into nonlinear controller design and apply it in the real robot system.
Keywords : State estimation, gaussian filter, Particle Filter, nonparametric method, nonlinear control.

Table des matières

1

Introduction

The foundation of understanding the dynamic phenomenon is to estimate its true state. State estimation problem is actually a filtering problem which uses the history of observation sequence to estimate the underlying hidden state sequence. It could be widely used for a large number of real applications in science and engineering, such as economics, biostatistics, stochastic signal processing, robot localization and navigation, etc. Hence, accurate and efficient state estimation is very crucial to analyze the properties in the complicated dynamic systems.

During the past decades, there have been a great number of filtering algorithms for state estimation problem. The most classic algorithm is a class of gaussian filters which estimate the posterior distribution by a gaussian distribution. Kalman Filter [Kalman, 1960] is the best-known optimal filter for linear and gaussian systems. But Kalman Filter is relatively theoretical since most real systems are nonlinear and/or nongaussian. Hence, to efficiently solve state estimation problems in practice, Extended Kalman Filter [Jazwinski, 1970] was developed by linearizing the nonlinear system to satisfy the Kalman Filter Framework. But due to the fact that Extended Kalman Filter approximates the posterior distribution as a gaussian distribution, it works terribly when the true posterior distribution is highly nongaussian. Gaussian Sum Filter[Daniel *et al.*, 1972] applies a mixture of gaussian distributions to approximate the posterior distribution, which works like a number of Extended Kalman Filters in parallel. However, both Extended Kalman Filter and Gaussian Sum Filter have to calculate the Jacobian matrices with high computational cost. Therefore, the derivative-free Unscented Kalman Filter [Simon *et al.*, 1995; Eric *et al.*, 2000] was designed using deterministic sigma points directly to approximate the posterior density in Kalman Filter framework. Even though there has been a lot of application using this gaussian filter class, the gaussian assumption will deteriorate the performance when the true distribution is totally nongaussian.

Monte Carlo sampling methods are the alternative algorithms to approximate the nongaussian posterior distribution by stochastic sampling. Sequential Monte Carlo Sampling method [Jun *et al.*, 1998; Arnaud *et al.*, 2000; Olivier *et al.*, 2007] is the one of the most efficient algorithm for filtering problem, and the resulting filter is a class of Particle Filter. Sequential Importance Sampling Particle filter [Zhe, 2003] is a popular framework to estimate the important region of posterior distribution recursively. However, weight degeneracy problem in this

algorithm would heavily affect the approximation performance since the variance of the particle weights will be very large after a few iterations [Zhe, 2003; Caglar *et al.*, 2011; Fred *et al.*, 2011; Samarjit, 2010]. Sampling Importance Resampling Particle Filter [Arnaud *et al.*, 2000; Deok-Jin, 2005] alleviates weight degeneracy problem with a Resampling step after Importance Sampling at each time step. But this Particle Filter just reduces the degeneracy problem rather than solving it due to the fact that Resampling step actually introduces high correlation between particles after a few iterations through replicating the particles with high important weight, which is the root why this Particle Filter framework works poorly in high dimensional systems [Fred *et al.*, 2011].

As mentioned before, there still exists serval difficulties in filtering problem, especially when the dynamic system is high-dimensional and nongaussian. Bayesian Nonparametric Technique is a promising strategy that could tackle the estimation problem for more general dynamic systems. Therefore, the aims of this proposal is to apply Nonparametric Technique to estimation problem with high accuracy as well as acceptable computational cost. Firstly, we attempt to take advantage of sequential Monte Carlo and MCMC to improve the estimation accuracy by combining them together with adaptive scheme. Secondly, to overcome the underlying weight degeneracy problem in the Particle Filter, we design a good proposal distribution using nonparametric methods. Thirdly, because dynamic state space model is unknown in the real world, we try to develop a filtering framework for nonparametric dynamic system so that the proposed algorithms could be applied for a more general state estimation problem. Then, we propose to estimate the states of nonparametric dynamic model with smoothing rather filtering since the performance of smoothing is better [Stuart *et al.*, 2010; Carlos *et al.*, 2010]. Finally, control problem for nonlinear systems is actually the final goal [Nadine *et al.*, 2007]. In this context, we plan to incorporate the nonparametric particle filter into a nonlinear controller in order to increase the control robustness [Villiers *et al.*, 2011; Stahl *et al.*, 2011].

Here is the outline of this research proposal. In Chapter 2, probabilistic inference tasks in temporal dynamic system is briefly introduced with a general state space model, then we deviate how to infer the target density using recursive bayesian estimation for filtering. Chapter 3 is the foundation of Bayesian Filtering theory starting from Kalman Filter to Particle Filter. After we illustrate the shortcomings of the existing filters, Chapter 4 will be used to propose my objectives and goals. Then the current research works following research goals are shown in Chapter 5. In Chapter 6, I show my work plan and time schedule. Finally, there is a conclusion for this proposal in Chapter 7.

2

Probabilistic Inference in Temporal Dynamic System

Probabilistic inference involves such a class of problems : according to the history of observable information, we attempt to find an efficient method to estimate the present, predict the future and assess the past. There are a great number of its applications which are widely used in industry, transportation, biomedicine, communication and so on. For example, the task is to evaluate the blood sugar level for a patient. By using the past blood sugar information, the nurse try to interpret the present level and predict its change in order to set the future medical plan. However, the blood sugar changes rapidly over time. Hence, before estimating the current and future values, we have to model the dynamic phenomena first.

2.1 Modeling for Temporal Dynamic System

Without loss of generality, the following dynamic state space model of a nonlinear system is considered for inference problem :

$$\mathbf{x}_t = \mathbf{a}(\mathbf{x}_{t-1}, \mathbf{w}_{t-1}) \tag{2.1}$$

$$\mathbf{y}_t = \mathbf{b}(\mathbf{x}_t, \mathbf{v}_t) \tag{2.2}$$

where at time t, $\mathbf{x}_t$ is the hidden system state vector, $\mathbf{w}_t$ is the system noise vector, $\mathbf{y}_t$ is the observation vector, $\mathbf{v}_t$ is the observation noise vector.

In the state equation (2.1), current state vector $\mathbf{x}_t$ is generated by a nonlinear function $\mathbf{a}$ of previous state $\mathbf{x}_{t-1}$ and system noise vector $\mathbf{w}_{t-1}$. In the observation equation (2.2), current observation vector $\mathbf{y}_t$ is obtained by a nonlinear function $\mathbf{b}$ of current state $\mathbf{x}_t$ and observation noise vector $\mathbf{v}_t$. Hence, by assuming the probability distribution of system noise $\mathbf{w}_t$ and observation noise $\mathbf{v}_t$, the state and observation equation actually represent transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$ respectively.

Additionally, we assume that state transition and observation process satisfy first-order Markovian, i.e., the current state vector $\mathbf{x}_t$ only depends on the last state vector $\mathbf{x}_{t-1}$, which means that the conditional probability of current state $\mathbf{x}_t$ given all the past states $\mathbf{x}_{0:t-1} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{t-1}\}$ is equal to the transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$; the current observation vector $\mathbf{y}_t$ is only determined by the current state vector $\mathbf{x}_t$, which illustrates that the conditional probability of current observation $\mathbf{y}_t$ given all the states $\mathbf{x}_{0:t}$ and all the past observations $\mathbf{y}_{0:t-1}$ is equal to the likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$.

According to the Markovian assumption, the state space model is actually the functional form of the hidden markov model, which consists of initial density $p(\mathbf{x}_0)$, transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$.

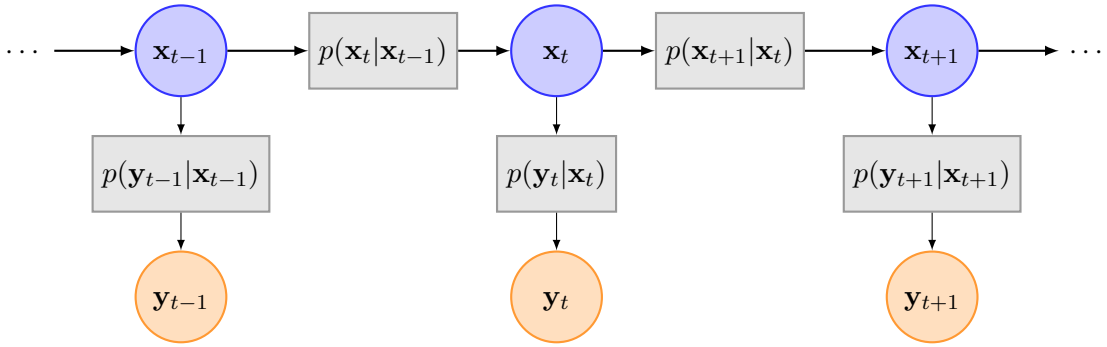

FIGURE 2.1: Hidden Markov Model.

2.2 Inference in Temporal Dynamic System

After setting up the general temporal model, we could classify the inference task into four parts [Stuart *et al.*, 2010; Heijden *et al.*, 2004] :

Filtering the task is to evaluate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{0:t})$. Filtering is also called state estimation.

Prediction the task is to evaluate the posterior distribution $p(\mathbf{x}_{t+k}|\mathbf{y}_{0:t})$ for $k > 0$ over the future state.

Smoothing the task is to evaluate the posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{0:t})$ for $0 \leq k < t$ over the past state. Smoothing offers a better estimation of the state than filtering since it incorporates more observation information, but it's an off-line method.

Most likely explanation the task is to find the most likely state sequence given the observation to date. The method to tackle this problem is the so called Viterbi algorithm which is also an off-line technique.

Additionally, there is a modeling task - **learning**. The goal is to learn the state space model or hidden markov model using the observation sequence.

Here we mainly focus on the filtering task. There are at least two main reasons. Firstly, the states themselves in the real world are very important for evaluating the dynamic systems.

For example, a mobile robot infers where it is by using filtering techniques to make the further path plan. Secondly, the states are estimated online and treated as a feedback to the nonlinear controller in order to improve the transitional and steady performance of close loop control system. For example, the position and orientation states in a mobile robot are estimated to control the transitional and rational velocity so that the tracking error could be reduced. Therefore, it is of importance to explore the different kinds of filtering algorithms for the real engineering applications.

2.3 Recursive Bayesian Filtering

To be able to efficiently calculate the posterior distribution of the hidden state sequence, the first intuition is to use Bayes Rule and Markov assumption to transform $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ into the following form that could be computed recursively.

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{p(\mathbf{y}_{0:t}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_t, \mathbf{y}_{0:t-1}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_t, \mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t, \mathbf{y}_{0:t-1}, \mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{y}_{0:t-1})p(\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{0:t-1}, \mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})p(\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})p(\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})} \end{aligned} \tag{2.3}$$

where

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1} \tag{2.4}$$

$$p(\mathbf{y}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t \tag{2.5}$$

the equation (2.4) represents a one-step prediction of the next state and the equation (2.5) updates the prediction with new observation.

Generally, the calculation of the integrations in the equation (2.4) and equation (2.5) determine whether the filtering problem could be solved or not, and the integrations largely depend on the characteristics of dynamic systems which classify the existing filtering methods. For the linear and gaussian systems, all the probability distributions mentioned are multi-variable gauss distributions obtained by linear transmission in the state-space model. Hence, there exists an integrable optimal form of $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, which is the well-known Kalman Filter. However, for most applications in the real world, the dynamic systems are nonlinear and non-gaussian, which leads to the fact that the computation of equation (2.4) and equation (2.5) are not feasible. Hence, the numerical approximation methods have been widely developed for the past decades. The following chapter will introduce serval typical filtering algorithms for state estimation problem in the bayesian framework.

3

Background and State of the Art

3.1 Gaussian Filter

Stochastic filtering refers to an issue for estimating the latent variables from the observable phenomena using a stochastic model. Since it was developed in early 1940s, filtering has been one of the most practical tools in the science and engineering field.

One of the best-known filters is Kalman Filter [Kalman, 1960], which is an optimal minimum mean squared error estimator for linear and gaussian systems. However, due to the fact that most of the dynamic systems in the real world are nonlinear and nongaussian, estimation analysis by Kalman Filter is intractable. In order to solve estimation problems in practice, from mid 1960s, a great number of extensions have been investigated to approximately estimate the hidden states by modifying the Kalman Filter framework. The classic method is Extended Kalman Filter [Jazwinski, 1970], which linearizes the nonlinear system using Taylor expansion to satisfy the Kalman Filter Framework. Even though it has been successfully used in many nonlinear problems, Extended Kalman Filter is limited by assuming that the true posterior density is a gaussian distribution. Especially when the true posterior distribution is totally different from a gaussian distribution, filtering performance by Extended Kalman Filter is heavily distorted. Gaussian sum filter[Daniel *et al.*, 1972] attempts to use a gaussian mixture instead of one gaussian distribution to approximate the arbitrary posterior distribution, which is actually designed as a number of Extended Kalman Filters in parallel. However, due to the fact that it applies EKF framework, the Jacobian matrices have to be calculated at each iteration which greatly increases the computational cost. Therefore, a lot of attention has been paid recently to the derivative-free sampling methods, which draws samples from the target distribution to avoid Jacobian matrices computation. One of the representative deterministic approximation algorithms is Unscented Kalman Filter [Simon *et al.*, 1995; Eric *et al.*, 2000]. Applying so-called Unscented Transformation, Unscented Kalman Filter directly uses sigma points to estimate the latent states in general Kalman Filter framework.

No matter which methods mentioned before, they all estimate the posterior distribution as a gaussian distribution or mixture. Hence they are a class of so called gaussian filters. In this section, we introduce these gaussian filters starting from the basic Kalman Filter.

3.1.1 Kalman Filter

Being considered as the simplified version of the general nonlinear system, the linear and gaussian system is represented as follows :

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_{t-1} \tag{3.1}$$

$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \tag{3.2}$$

where the nonlinear function $\mathbf{a}(\mathbf{x}_{t-1}, \mathbf{w}_{t-1})$ and $\mathbf{b}(\mathbf{x}_t, \mathbf{v}_t)$ in the general model in the equation (2.1) and (2.2) are transformed into $\mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_{t-1}$ and $\mathbf{H}\mathbf{x}_t + \mathbf{v}_t$ respectively. $\mathbf{F}$ is the constant linear transition matrix and $\mathbf{H}$ is the constant observation matrix. Both system noise $\mathbf{w}_t$ and observation noise $\mathbf{v}_t$ are white noise processes with known covariance, i.e., $\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{Q})$, $\mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R})$, where $\mathbf{Q} = E\left[\mathbf{w}_t \mathbf{w}_t^T\right]$ and $\mathbf{R} = E\left[\mathbf{v}_t \mathbf{v}_t^T\right]$.

Due to the fact that gaussian distribution is closed under the linear transformation, the transition probability is $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim N(\mathbf{F}\mathbf{x}_{t-1}, \mathbf{Q})$, the likelihood density is $p(\mathbf{y}_t|\mathbf{x}_t) \sim N(\mathbf{H}\mathbf{x}_t, \mathbf{R})$. It sequentially ensures that the integration in the equation (2.4) and equation (2.5) will produce gaussian distributions. Therefore, the posterior density will be gaussian according to the equation (2.3) in the recursive bayesian filtering. This closed-form filter is the well known Kalman Filter.

Kalman Filter is composed of an iterative predict-update process [Greg *et al.*, 2001; Thacker *et al.*, 2006; Tristan, 2010; Nathan, 2003]. Here is the general framework :

In the predict step, the estimated state from the previous time is projected to produce a current state estimation, which is also called the a priori estimation due to the fact that current observation is not considered.

– Priori state estimation

$$\hat{\mathbf{x}}_t' = \mathbf{F}\hat{\mathbf{x}}_{t-1} \tag{3.3}$$

– Priori covariance estimation

$$\mathbf{P}_t' = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{Q} \tag{3.4}$$

According to the estimated state $\hat{\mathbf{x}}_{t-1}$, Kalman Filter makes one step prediction of the current state and propagates the estimated covariance $\mathbf{P}_{t-1}$ using transition probability distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim N(\mathbf{F}\mathbf{x}_{t-1}, \mathbf{Q})$ in the state equation.

In the update step, observation information is integrated with the prediction to correct the priori estimation to obtain a posteriori estimation.

– Kalman Gain

$$\mathbf{K}_t = \mathbf{P}_t'\mathbf{H}^T(\mathbf{H}\mathbf{P}_t'\mathbf{H}^T + \mathbf{R})^{-1} \tag{3.5}$$

– Posteriori state estimation

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t' + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_t') \tag{3.6}$$

– Posteriori covariance estimation

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t' \tag{3.7}$$

The posterior estimated state $\hat{\mathbf{x}}_t$ is expressed as prior estimated state $\hat{\mathbf{x}}_t'$ plus the correction item based on the difference between predict observation $\mathbf{H}\hat{\mathbf{x}}_t'$ and actual observation $\mathbf{y}_t$. The posteriori covariance estimation could be represented as prior estimated covariance $\mathbf{P}_t'$ minus

the covariance correction $\mathbf{K}_t\mathbf{H}\mathbf{P}'_t$ when the current observation is available. Additionally, the Kalman Gain is obtained using the mean square error measure in order to minimize the covariance $\mathbf{P}_t$. The whole update step illustrates that the estimated state would be more accurate with smaller covariance after incorporating the observation.

Even though Kalman Filter is an optimal algorithm with an elegant recursive form for linear and gaussian systems, it is still limited since most of the real systems are nonlinear and/or nongaussian. In order to apply the Kalman Filter framework in practice, a so called Extended Kalman filter is designed by relaxing the linear assumption in Kalman Filter.

3.1.2 Extended Kalman Filter

From the perspective of bayesian filtering, integration of equation (2.4) and equation (2.5) for real systems are impossible to be calculated directly like Kalman Filter because the related probability distributions of states and observations are not standard guassian distributions. Hence, approximation methods are the essence of filtering problem for a nonlinear and/or nongaussian system. Extended Kalman filter is one of the well-known approximate estimators.

Without loss of generality, we consider here a simple class of the general nonlinear system in equation (2.1) and (2.2) with additive gaussian noise :

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{w}_{t-1} \tag{3.8}$$

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t \tag{3.9}$$

where the nonlinear function $\mathbf{a}(\mathbf{x}_{t-1}, \mathbf{w}_{t-1})$ and $\mathbf{b}(\mathbf{x}_t, \mathbf{v}_t)$ in the general model in the equation (2.1) and (2.2) are transformed into $\mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{w}_{t-1}$ and $\mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t$ respectively. $\mathbf{f}$ and $\mathbf{g}$ are nonlinear differentiable function. Both system noise $\mathbf{w}_t$ and observation noise $\mathbf{v}_t$ are white noise processes with known covariance, i.e., $\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{Q})$, $\mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R})$. Then the transition probability is $p(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim N(\mathbf{f}(\mathbf{x}_{t-1}), \mathbf{Q})$, the likelihood density is $p(\mathbf{y}_t|\mathbf{x}_t) \sim N(\mathbf{h}(\mathbf{x}_t), \mathbf{R})$.

The general idea of Extended Kalman Filter is to linearize the nonlinear system by using first-order taylor expansion to satisfy the linear assumption of Kalman filter, then the state estimation problem could be solved using Kalman Filter framework [Shoudong, 2010; Maria, 2004].

Here state equation is linearized at $\hat{\mathbf{x}}_{t-1}$,

$$\begin{aligned} \mathbf{x}_t &\approx \mathbf{f}(\hat{\mathbf{x}}_{t-1}) + \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}) + \mathbf{w}_{t-1} \\ &= \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}\mathbf{x}_{t-1} + \mathbf{f}(\hat{\mathbf{x}}_{t-1}) - \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}\hat{\mathbf{x}}_{t-1} + \mathbf{w}_{t-1} \end{aligned} \tag{3.10}$$

where $\nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}$ is the Jacobian of $\mathbf{f}$ at $\hat{\mathbf{x}}_{t-1}$. Then we could get the linearized state equation

$$\mathbf{x}_t \approx \mathbf{F}\mathbf{x}_{t-1} + \mathbf{U}_{t-1} + \mathbf{w}_{t-1} \tag{3.11}$$

where

$$\mathbf{F} = \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}} \tag{3.12}$$

$$\begin{aligned} \mathbf{U}_{t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1}) - \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}\hat{\mathbf{x}}_{t-1} \\ &= \mathbf{f}(\hat{\mathbf{x}}_{t-1}) - \mathbf{F}\hat{\mathbf{x}}_{t-1} \end{aligned} \tag{3.13}$$

observation equation is linearized at $\hat{\mathbf{x}}'_t$,

$$\mathbf{y}_t \approx \mathbf{h}(\hat{\mathbf{x}}'_t) + \nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t}(\mathbf{x}_t - \hat{\mathbf{x}}'_t) + \mathbf{v}_t$$

where $\nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t}$ is the Jacobian of $\mathbf{h}$ at $\hat{\mathbf{x}}'_t$. Then the linearized observation equation is transformed to the following form

$$\mathbf{y}_t - \mathbf{h}(\hat{\mathbf{x}}'_t) + \nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t}\hat{\mathbf{x}}'_t \approx \nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t}\mathbf{x}_t + \mathbf{v}_t$$

Let

$$\mathbf{H} = \nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t} \tag{3.14}$$

$$\mathbf{Y}_t = \mathbf{y}_t - \mathbf{h}(\hat{\mathbf{x}}'_t) + \mathbf{H}\hat{\mathbf{x}}'_t \tag{3.15}$$

Then the final linearized observation equation is as follows

$$\mathbf{Y}_t \approx \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \tag{3.16}$$

Now applying the linearized state space model into Kalman Filter Framework, Extended Kalman Filter is as follows :

In the predict step, the priori estimation at t is

$$\hat{\mathbf{x}}'_t = \mathbf{f}(\hat{\mathbf{x}}_{t-1}) \tag{3.17}$$

$$\mathbf{P}'_t = \nabla\mathbf{f}|_{\hat{\mathbf{x}}_{t-1}}\mathbf{P}_{t-1}\nabla\mathbf{f}^T|_{\hat{\mathbf{x}}_{t-1}} + \mathbf{Q} \tag{3.18}$$

In the update step, the posteriori estimation at t is

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}(\hat{\mathbf{x}}'_t)) \tag{3.19}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t})\mathbf{P}'_t \tag{3.20}$$

where Kalman Gain $\mathbf{K}_t$ is defined :

$$\mathbf{K}_t = \mathbf{P}'_t\nabla\mathbf{h}^T|_{\hat{\mathbf{x}}'_t}(\nabla\mathbf{h}|_{\hat{\mathbf{x}}'_t}\mathbf{P}'_t\nabla\mathbf{h}^T|_{\hat{\mathbf{x}}'_t} + \mathbf{R})^{-1} \tag{3.21}$$

Since Extended Kalman Filter relaxes the linear assumption in Kalman Filter, the state estimation problem for a class of nonlinear differentiable systems could be done in the real world. One of the most important applications is robot localization due to the fact that the kinematic model and observation model of mobile robot is known and differentiable, and also the elegant update form is suitable for real time implementation. For example, [Evgeni *et al.*, 2002] applies Extended Kalman Filter into a lawn mover to localize its position. Additionally, [Yibing *et al.*, 2005] applies Extended Kalman Filter in the Real-time freeway traffic state estimation problem, Speed and Rotor Position Estimation of Brushless DC Motor is designed using Extended Kalman Filter in [Bozo *et al.*, 2001].

However, there are two main drawbacks in Extended Kalman Filter. Firstly, Extended Kalman Filter keeps the underlying gaussian assumption. Hence, when the true posterior distribution is not close to a gaussian distribution, the filtering performance by Extended Kalman Filter is heavily distorted. The other shortcoming is the computation of Jacobian matrices in the Extended Kalman Filter framework, all the matrices have to be calculated at each iteration which greatly increases the computational cost.

Therefore, lots of attention has been paid recently to lighten its disadvantages to improve estimated performance. The first one could be done with Gaussian sum filter[Daniel *et al.*, 1972] which is designed using a gaussian mixture rather than one gaussian distribution in Extended Kalman Filter, and the second one could be solved by a deterministic sampling approximation methods to avoid Jacobian matrices computation. One of the most representative algorithms is Unscented Kalman Filter [Simon *et al.*, 1996]. From now on, I will introduce both of them one by one.

3.1.3 Gaussian Sum Filter

As mentioned above, Extended Kalman Filter works well when the posterior distribution of nonlinear system is approximately gaussian. However, when true probability distribution is nongaussian, for example, it is multimodal, the approximation performance by Extended Kalman Filter would be poor due to the loss of important information in the true density. For the multimodal distribution, Extended Kalman Filter works more like a maximum likelihood estimator rather than a minimum variance estimator [Daniel *et al.*, 1972], which leads that approximated distribution would mistakenly follow one of the peaks in the true density.

The motivation of Gaussian Sum Filter is to apply a weighted sum of gaussian densities as a filter to approximate the unknown posterior distribution due to the fact that any nongaussian distribution could be approximated by a sufficient number of gaussian mixture densities to some precise extent [Zhe, 2003].

Considering the same state space model in Extended Kalman Filter, Gaussian Sum Filter approximates the posterior distribution using a gaussian mixture. Hence, the gaussian sum representation could be considered as a convex combination of the output of Extended Kalman Filters performing in parallel, which makes each gaussian filter in Gaussian Sum Filter follow Extended Kalman Filter framework. Assuming that the initial distribution is a gaussian mixture, we could obtain the approximated posterior distribution recursively by the following gaussian sum filter [Daniel *et al.*, 1972; Anderson *et al.*, 1979; Jayesh *et al.*, 2003].

In the predict step, we suppose $p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$ is expressed as a gaussian mixture

$$p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1}) \sim \sum_{i=1}^{G} w_{(t-1)i} N(\hat{\mathbf{x}}_{(t-1)i}, \mathbf{P}_{(t-1)i}) \tag{3.22}$$

According to equation (2.4), $p(\mathbf{x}_t|\mathbf{y}_{0:t-1})$ approaches a gaussian sum

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \sim \sum_{i=1}^{G} \bar{w}_{ti} N(\hat{\mathbf{x}}'_{ti}, \mathbf{P}'_{ti}) \tag{3.23}$$

as $\mathbf{P}_{(t-1)i} \to 0$ for i=1,..., G, where $\hat{\mathbf{x}}'_{ti}, \mathbf{P}'_{ti}$ could be calculated using Extended Kalman Filter prediction framework :

$$\bar{w}_{ti} = w_{(t-1)i} \tag{3.24}$$

$$\hat{\mathbf{x}}'_{ti} = \mathbf{f}(\hat{\mathbf{x}}_{(t-1)i}) \tag{3.25}$$

$$\mathbf{P}'_{ti} = \nabla \mathbf{f}|_{\hat{\mathbf{x}}_{(t-1)i}} \mathbf{P}_{(t-1)i} \nabla \mathbf{f}^T|_{\hat{\mathbf{x}}_{(t-1)i}} + \mathbf{Q} \tag{3.26}$$

In the update step, with $p(\mathbf{x}_t|\mathbf{y}_{0:t-1})$ shown as a gaussian sum

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \sim \sum_{i=1}^{G} \bar{w}_{ti} N(\hat{\mathbf{x}}'_{ti}, \mathbf{P}'_{ti}) \tag{3.27}$$

the posterior distribution could be approximated as a gaussian mixture using equation (2.3) after receiving the observations

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) \sim \sum_{i=1}^{G} w_{ti} N(\hat{\mathbf{x}}_{ti}, \mathbf{P}_{ti}) \tag{3.28}$$

where $\hat{\mathbf{x}}_{ti}, \mathbf{P}_{ti}$ could be calculated using Extended Kalman Filter update framework :

$$\hat{\mathbf{x}}_{ti} = \hat{\mathbf{x}}'_{ti} + \mathbf{K}_{ti}(\mathbf{y}_t - \mathbf{h}(\hat{\mathbf{x}}'_{ti})) \tag{3.29}$$

$$\mathbf{P}_{ti} = (\mathbf{I} - \mathbf{K}_{ti} \nabla \mathbf{h}|_{\hat{\mathbf{x}}'_{ti}}) \mathbf{P}'_{ti} \tag{3.30}$$

$$\mathbf{K}_{ti} = \mathbf{P}'_{ti} \nabla \mathbf{h}^T|_{\hat{\mathbf{x}}'_{ti}} (\nabla \mathbf{h}|_{\hat{\mathbf{x}}'_{ti}} \mathbf{P}'_{ti} \nabla \mathbf{h}^T|_{\hat{\mathbf{x}}'_{ti}} + \mathbf{R})^{-1} \tag{3.31}$$

and the weight could be updated as follows [Anderson *et al.*, 1979] :

$$w_{ti} = \frac{\bar{w}_{ti} \alpha_{ti}}{\sum_{i=1}^{G} \bar{w}_{ti} \alpha_{ti}} \tag{3.32}$$

$$\alpha_{ti} = N(\mathbf{y}_t; \mathbf{h}(\hat{\mathbf{x}}'_{ti}), \nabla \mathbf{h}|_{\hat{\mathbf{x}}'_{ti}} \mathbf{P}'_{ti} \nabla \mathbf{h}^T|_{\hat{\mathbf{x}}'_{ti}} + \mathbf{R}) \tag{3.33}$$

Here, a multi-equilibrium signal process in [Kazufumi *et al.*, 2000] is considered in order to illustrate the efficiency of Gaussian Sum Filter for the multimodal question compared with Extended Kalman Filter. The state space model of one-dimensional signal process is as follows :

$$x_t = x_{t-1} + 5T x_{t-1}(1 - x_{t-1}^2) + w_{t-1} \tag{3.34}$$

$$y_t = T(x_t - 0.05)^2 + v_t \tag{3.35}$$

where the system and observation noise are $w_t \sim N(0, b^2 T)$, $v_t \sim N(0, d^2 T)$ respectively. In this model, there are three equilibria -1, 0 and 1 where -1 and 1 are stable equilibria. Therefore, depending on initial state choice and stochastic noise, the system state would be eventually distributed around -1 or 1.

Here we choose time interval $T = 0.005$, the ending time is 5, initial state is $x_0 = 0$, $b = 0.5$, $d = 0.1$. Additionally, we use two gauss distributions to construct the gaussian sum filter, and the initial distributions are $N(-1, 2)$, $N(1, 2)$. The initial weights for these two normal distributions are both 0.5.

From the simulation, the hidden state finally reaches around the stable state 1 and two time points $t = 100T, 1000T$ are chosen in figure 3.1 to illustrate how the gaussian sum filter works. The approximated distributions at these time points are illustrated in figure 3.2, and figure 3.3 presents the change of the weights in gaussian sum filter.

At $t = 10T$, the state is around 0 in figure 3.1. At that time point, the mixed degree of two gaussian distributions is relatively large in figure 3.2, which leads that the weights of them are almost half and half in figure 3.3. Moreover, due to the fact that the state is definitely around

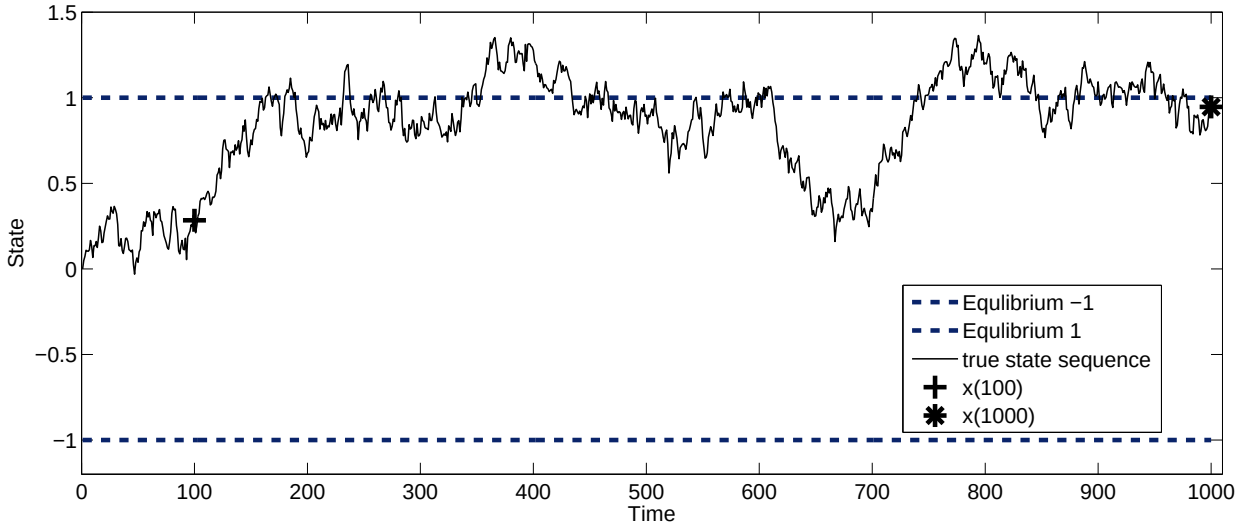

FIGURE 3.1: The system state over time

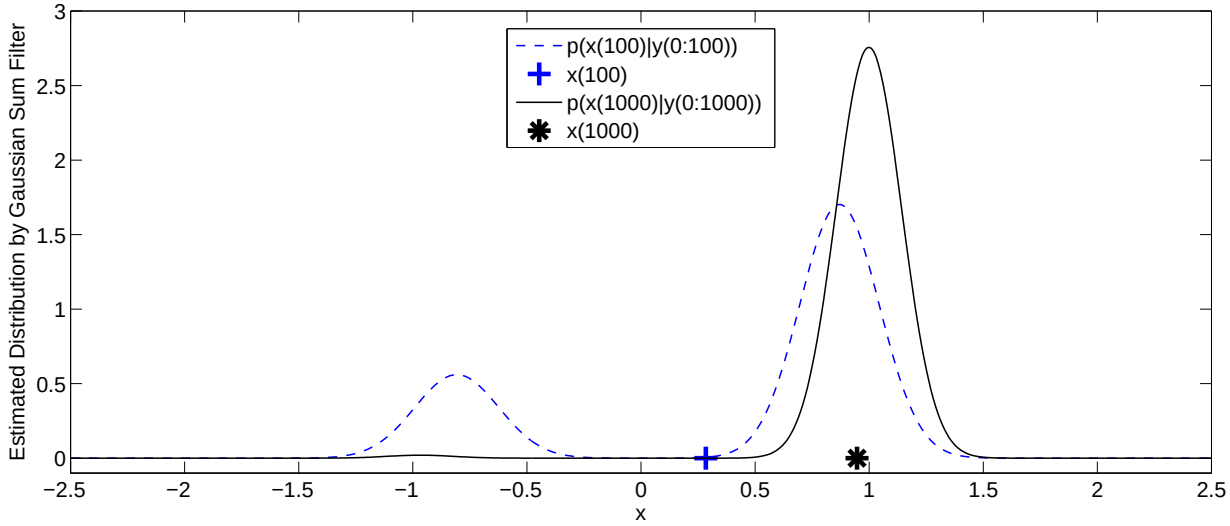

FIGURE 3.2: Approximated distribution at $t = 100T, 1000T$ by Gaussian Sum Filter

1 at $t = 1000T$ in figure 3.1, the gaussian density that is from initial distribution $N(1, 2)$ plays a leading role in in figure 3.2, whose weight is almost 1 in in figure 3.3.

Finally, in order to illustrate Gaussian sum filter performs better than EKF for multimodal distribution, we compare the approximated posterior density at $t = 100T$, $1000T$, which are shown in figure 3.4 and 3.5. In both figures, the state is mistakenly estimated as -1 by Extended Kalman Filter. On the contrary, Gaussian Sum Filter gets the correct state estimation in both

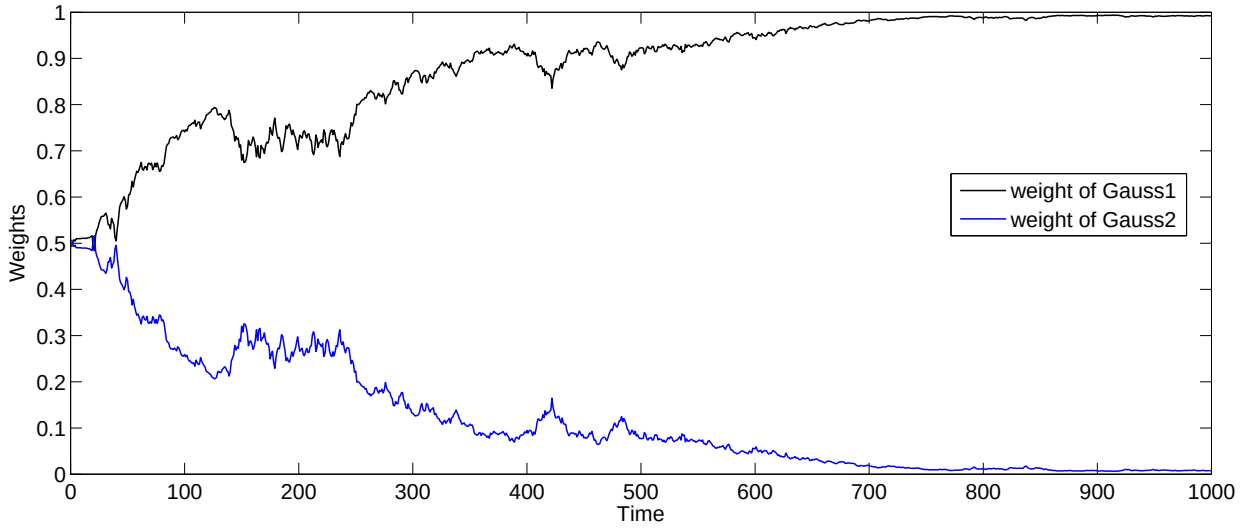

FIGURE 3.3: The change of two gaussian distributions' weight over time

time points because it is the multi-Extended Kalman Filters parallel operation through a weight sum to obtain more distribution information.

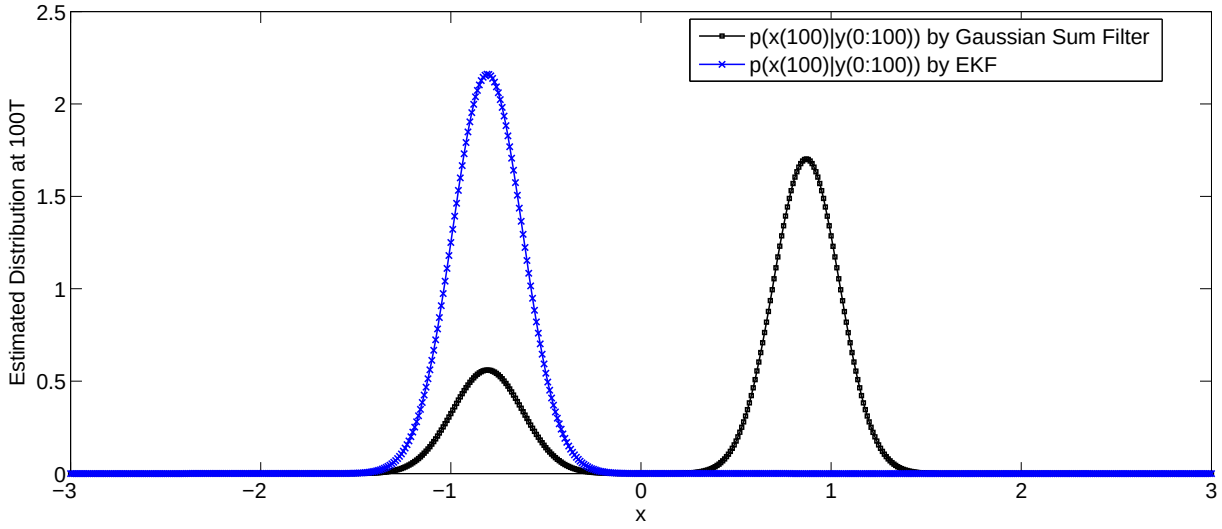

FIGURE 3.4: Approximated distribution at $t = 100T$

Gaussian Sum Filter successfully applies a gaussian mixture to approximate the complicated posterior distribution. However, the mechanism Gaussian Sum Filter is to implement the algorithm as a number of Extended Kalman Filters, therefore, the computation of Jacobian matrices would increase very fast, especially when the state space is high dimensional.

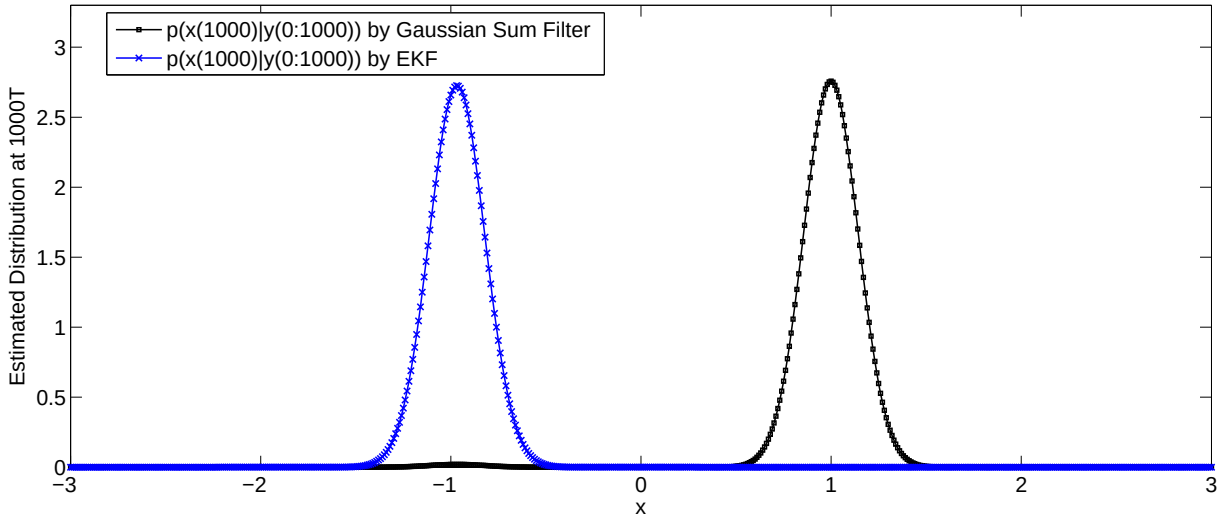

FIGURE 3.5: Approximated distribution at $t = 1000T$

As mentioned before, a so called Unscented Kalman Filter provides us a derivative-free algorithm in which the posterior distribution could be estimated by a couple of deterministic sigma points.

3.1.4 Unscented Kalman Filter

Even though Extended Kalman filter keeps the computationally efficient online form of Kalman Filter, it suffers serval limitations [Simon *et al.*, 1995; Xionga *et al.*, 2006; Banani *et al.*, 2007]. The predicted mean in the Extended Kalman filter is equal to the prior mean mapping through the nonlinear function. It means that Extended Kalman filter does not consider the affection of noise distribution, which leads that the filtering performance is highly unstable. What is more, the computation of Jacobian matrices is not trivial in most of real applications, especially in high dimensional systems. Hence, in order to improve the accuracy of filtering efficiently, an Unscented Kalman Filter is developed using unscented transformation in the General Kalman Filter framework [Simon *et al.*, 1995, 1996, 2004; Xionga *et al.*, 2006; Banani *et al.*, 2007].

Unscented Transformation is a deterministic sampling method, which applies sigma points with fixed parameters to propagate the information of data [Zhe, 2003; Eric *et al.*, 2000]. Considering the following general nonlinear equation :

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{3.36}$$

the goal is to predict the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_y$ of random variable $\mathbf{y}$ using the mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_x$ of random variable $\mathbf{x}$. The specific procedure is as follows [Simon *et al.*, 2004; Eric *et al.*, 2000] :

– Step One : to form a matrix χ of $2L + 1$ sigma vectors χ_i

$$\chi_0 = \bar{\mathbf{x}} \tag{3.37}$$

$$\chi_i = \bar{\mathbf{x}} + (\sqrt{(L+\lambda)\mathbf{P}_x})_i, i = 1, \cdots, L \tag{3.38}$$

$$\chi_i = \bar{\mathbf{x}} - (\sqrt{(L+\lambda)\mathbf{P}_x})_{i-L}, i = L+1, \cdots, 2L \tag{3.39}$$

where L is the dimension of $\mathbf{x}$, $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter, α is set to a small positive constant and κ is set to 0 or $3 - L$ generally.

– Step Two : to propagate $2L+1$ χ_i through the nonlinear function :

$$\mathcal{Y}_i = \mathbf{g}(\chi_i) \tag{3.40}$$

and then, to use the new $2L+1$ vectors $\mathcal{Y}_i$ to calculate the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_y$:

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i \tag{3.41}$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2L} W_i^{(c)} \{\mathcal{Y}_i - \bar{\mathbf{y}}\}\{\mathcal{Y}_i - \bar{\mathbf{y}}\}^T \tag{3.42}$$

where the weights are given by :

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{3.43}$$

$$W_0^{(c)} = \lambda/(L+\lambda) + (1 - \alpha^2 + \beta) \tag{3.44}$$

$$W_i^{(m)} = W_i^{(c)} = 1/\{2(L+\lambda)\}, i = 1, \cdots, 2L \tag{3.45}$$

$\beta = 2$ is optimal for gaussian distribution.

Incorporating Unscented Transformation into general Kalman Filter framework, a derivative-free Unscented Kalman Filter is designed with deterministic sigma points instead of Jacobian computation. The intuition of Unscented Kalman Filter is following [Simon *et al.*, 1995] : *With a fixed number of parameters, it should be easier to approximate a gaussian distribution than it is to approximate an arbitrary nonlinear function.*

Let's consider the same nonlinear system in Extended Kalman Filter but there is no differential assumption for the nonlinear function, the Unscented Kalman Filter is as follows [Eric *et al.*, 2000; Rambabu *et al.*, 2008; Jouni *et al.*, 2007] :

– Initialization :

$$\mathbf{x}_0 \sim N(\hat{\mathbf{x}}_0, \mathbf{P}_0) \tag{3.46}$$

$$\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{Q}) \tag{3.47}$$

$$\mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R}) \tag{3.48}$$

Setting :

$$\hat{\mathbf{x}}_0^a = \begin{bmatrix} \hat{\mathbf{x}}_0^T & \mathbf{0} & \mathbf{0} \end{bmatrix}^T \tag{3.49}$$

$$\mathbf{P}_0^a = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{3.50}$$

For $t = 1, 2, ...$

– Sigma points calculation :

$$\begin{aligned}\chi^a_{t-1} &= \left[\hat{\mathbf{x}}^a_{t-1} \quad \hat{\mathbf{x}}^a_{t-1} + \sqrt{(L+\lambda)\mathbf{P}^a_{t-1}} \quad \hat{\mathbf{x}}^a_{t-1} - \sqrt{(L+\lambda)\mathbf{P}^a_{t-1}}\right] \\ &= \left[(\chi^x_{t-1})^T \quad (\chi^w_{t-1})^T \quad (\chi^v_{t-1})^T\right]^T \end{aligned} \tag{3.51}$$

– Prediction :

$$\chi^x_{t|t-1} = \mathbf{f}(\chi^x_{t-1}) + \chi^w_{t-1} \tag{3.52}$$

$$\hat{\mathbf{x}}'_t = \sum_{i=0}^{2L} W_i^{(m)} \chi^x_{i,t|t-1} \tag{3.53}$$

$$\mathbf{P}'_t = \sum_{i=0}^{2L} W_i^{(c)} (\chi^x_{i,t|t-1} - \hat{\mathbf{x}}'_t)(\chi^x_{i,t|t-1} - \hat{\mathbf{x}}'_t)^T \tag{3.54}$$

$$\mathcal{Y}_{t|t-1} = \mathbf{h}(\chi^x_{t|t-1}) + \chi^v_{t-1} \tag{3.55}$$

$$\hat{\mathbf{y}}'_t = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,t|t-1} \tag{3.56}$$

– Update :

$$\mathbf{P}_{yy} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Y}_{i,t|t-1} - \hat{\mathbf{y}}'_t)(\mathcal{Y}_{i,t|t-1} - \hat{\mathbf{y}}'_t)^T \tag{3.57}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2L} W_i^{(c)} (\chi^x_{i,t|t-1} - \hat{\mathbf{x}}'_t)(\mathcal{Y}_{i,t|t-1} - \hat{\mathbf{y}}'_t)^T \tag{3.58}$$

$$\mathbf{K}_t = \mathbf{P}_{xy}\mathbf{P}^{-1}_{yy} \tag{3.59}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}'_t) \tag{3.60}$$

$$\mathbf{P}_t = \mathbf{P}'_t - \mathbf{K}_t\mathbf{P}_{yy}\mathbf{K}^T_t \tag{3.61}$$

Here we considering the following Univariate Nonstationary Growth Model in [Jouni *et al.*, 2007] as an example to compare Unscented Kalman Filter and Extended Kalman Filter :

$$x_t = 0.5x_{t-1} + 25x_{t-1}(1 + x^2_{t-1})^{-1} + 8\cos(1.2t) + w_{t-1} \tag{3.62}$$

$$y_t = 0.05x^2_t + v_t \tag{3.63}$$

where the system and observation noise are $w_t \sim N(0,1)$, $v_t \sim N(0,1)$ respectively, the initial state distribution for both Extended Kalman Filter and Unscented Kalman Filter is $x_0 \sim N(0,1)$. Additionally, the time interval is 0.01, the terminal time is 0.5, and α =1, $\beta = 2$, κ=2.

The simulation results in figure 3.6 and table 3.1 show that the estimated state by Unscented Kalman Filter is closer to the real state than the estimated state by Extended Kalman Filter because Unscented Kalman Filter gradually estimates the state using sigma points with noise consideration. However, we should notice that Unscented Kalman Filter is far way too perfect because it belongs to general Kalman Framework which means that it keeps the gaussian assumption for posterior distribution.

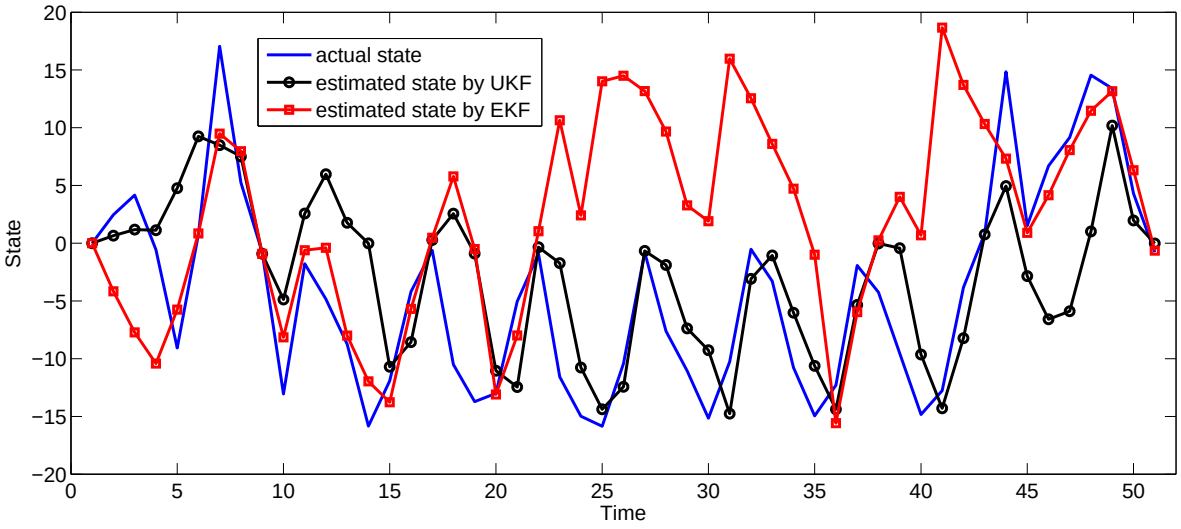

FIGURE 3.6: The state estimation by UKF and EKF over time

statistics	Error by UKF	Error by EKF
mean	-1.1881	-6.2913
variance	47.5356	108.4723

TABLE 3.1: Estimated error statistics

Unscented Kalman Filter effectively evaluates the posterior distribution through the sigma points propagation without any analytic differentiation operartion in Extended Kalman Filter. It captures the posterior mean and covariance to third order Taylor expansion instead of first order in Extended Kalman Filter [Eric *et al.*, 2000], where they have the same asymptotic complexity. However, Unscented Kalman Filter belongs to gaussian filter which means that it approximates the posterior distribution as a gaussian density. When the true distribution is highly nongaussian, the Unscented Kalman Filter would perform extremely poor.

3.1.5 Summary

Since R.E.Kalman published the distinguished paper to provide a recursive solution for discrete linear filtering problem [Kalman, 1960], Kalman Filter had been widely used in different kinds of tracking and control applications. By minimizing the mean squared error (MSE), Kalman Filter efficiently estimates the hidden states of the linear and gaussian systems. However, Kalman Filter is considered as a more theoretical result since linear and gaussian assumption is actually too strong to apply for most of dynamic systems in the real world.

Many extensions have been designed for nonlinear and/or nongaussian systems during the past decades. The most famous one is Extended Kalman Filter, which linearizes the nonlinear system to use the classic Kalman Framework. However, Extended Kalman Filter approximates the true probability density using a gaussian distribution which dramatically deteriorates the

estimation performance. Hence Gaussian Sum Filter works as a number of Extended Kalman Filters, which applies a weighted sum of gaussian mixture densities as a posterior approximation for nongaussian systems. But the computation of Jacobian matrices for both Extended Kalman Filter and Gaussian Sum Filter are not trivial in practice. Hence, an Unscented Kalman Filter is developed using unscented transformation to improve the estimated accuracy efficiently.

One of the primary advantage of this Gaussian Filter class is computational : the update requires time polynomial in the dimensionality of the state space [Thrun *et al.*, 2005]. However, this kind of filter will give rise to arbitrarily poor performance when the target distribution is nongaussian since they keep the underlying gaussian assumption of the true posterior distribution.

Hence more and more researchers have paid attention to the strategy which could deal with the general nonlinear and nongaussian systems. Monte Carlo methods in the next section have played a leading role by estimating the true posterior distribution with a collection of random samples.

3.2 Monte Carlo Sampling Methods

All the filters mentioned previously are based on Kalman Filter framework to estimate the hidden state in nonlinear system. However, the underlying gaussian assumption limits their application in the real world. In order to deal with more general nonlinear systems which don't put strong constraints on the posterior behavior, Monte carlo methods have been developed to approximate the probability distribution by producing a set of stochastic samples. In this Chapter, we will introduce two popular Monte Carlo methods : Importance Sampling and Markov Chain Monte Carlo. Both of them would be used as the foundation to construct a class of particle filters and future work.

3.2.1 Monte Carlo Principle

The core of Monte Carlo methods is to draw a set of samples $\{x^{(i)}\}_{i=1}^{N}$ from a target density $p(x)$ defined on a state space. Using these samples, the target density could be approximated with the empirical point-mass function as follows [Christophe *et al.*, 2003] :

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x) \tag{3.64}$$

where $\delta_{x^{(i)}}(x)$ is the delta-dirac mass located at $x^{(i)}$. Moreover, numerical integration problem could also be transformed as an expectation calculation which could be also estimated with these samples :

$$\hat{E}(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \longrightarrow E(f) = \int f(x)p(x)dx \tag{3.65}$$

when $N \to \infty$, the estimation is converge to $E(f)$ by strong law of large numbers.

However, in general the target distribution $p(x)$ is not a simple combination of standard distributions, which means that it is almost impossible to sample from it directly. For example,

if the target distribution $p(x)$ is gaussian, the samples could be drawn easily from it. However, if it's a highly nongaussian unknown distribution, we could not sample from it directly. The Important Sampling techniques are designed to deal with this difficulty. By drawing samples from a proposal distribution, the target distribution could be approximated by a weight sum.

3.2.2 Important Sampling

In order to improve computational efficiency, important sampling aims to draw the samples from the important region of the target distribution [Zhe, 2003; Christopher , 2006]. Assuming the support of proposal distribution $q(x)$ covers the support of target distribution $p(x)$, let's reconsider the integration problem :

$$E(f) = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = \int f(x)w(x)q(x)dx \tag{3.66}$$

where $w(x) \triangleq \dfrac{p(x)}{q(x)}$ is the important weight.

Using N i.i.d. samples drawn from $q(x)$, important sampling provides a weight sum estimation of the integration :

$$\hat{E}(f) = \frac{1}{N}\sum_{i=1}^{N} w(x^{(i)})f(x^{(i)}) \tag{3.67}$$

In practice, the normalized part of target distribution is unknown, therefore, the weight importance is proportional to $p(x)/q(x)$. Generally, we normalized weight importance to obtain more practical important sampling method.

$$E(f) = \int f(x)p(x)dx = \frac{\int f(x)\frac{p(x)}{q(x)}q(x)dx}{\int \frac{p(x)}{q(x)}q(x)dx} = \frac{\int f(x)w(x)q(x)dx}{\int w(x)q(x)dx} \tag{3.68}$$

Integration estimation by normalized important sampling transforms :

$$\hat{E}(f) = \frac{\frac{1}{N}\sum_{i=1}^{N} w(x^{(i)})f(x^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} w(x^{(i)})} = \sum_{i=1}^{N} \tilde{w}(x^{(i)})f(x^{(i)}) \tag{3.69}$$

where $\tilde{w}(x^{(i)}) = \dfrac{w(x^{(i)})}{\sum_{i=1}^{N} w(x^{(i)})}$ is the normalized important weight.

Similarly, the approximated target distribution is obtained by Monte Carlo principle :

$$\hat{p}(x) = \sum_{i=1}^{N} \tilde{w}(x^{(i)})\delta_{x^{(i)}}(x) \tag{3.70}$$

Below a trivial example is considered to illustrate how important sampling works. The target distribution is a gaussian mixture $p(x) \sim 0.5N(-3,1) + 0.5N(1,2^2)$, we choose an easy-sampling proposal distribution $q(x) \sim 3.2N(1,4^2)$. Then we draw 30 samples from the

proposal distribution and calculate their corresponding weights. The sampling results are shown in figure 3.7. The circles are the samples we draw from the proposal distribution which just provide the possible values from target distribution, and then we calculate the weight of the samples (stars). The weight is actually proportional to the probability of the possible values in the target distribution. Hence Importance Sampling only uses the samples from proposal distribution and their weight distribution to successfully approximate the target distribution.

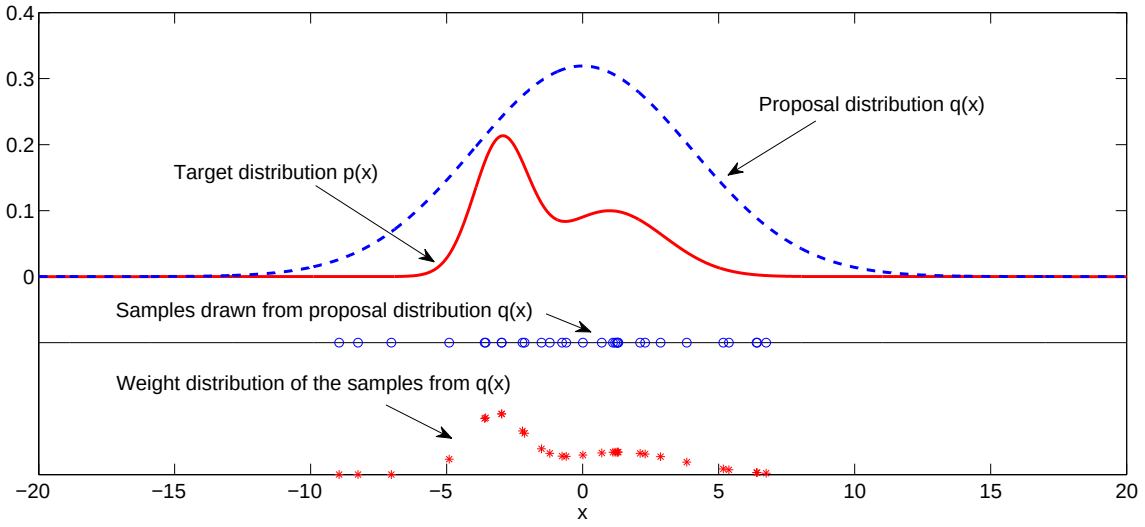

FIGURE 3.7: Estimated distribution by Important Sampling

However, there are two important issues in important sampling [MacKay , 1998]. Firstly, the proposal distribution is very crucial for this method. If it's not a good approximation of target distribution, the sampling performance would be unacceptable. For example, we attempt to obtain the samples in the high probability region of the target distribution. However, Importance Sample just allows us sample from the high probability region of the proposal distribution. Hence, if the high probability region of the proposal distribution is the low probability region of the target distribution. The sampling process from proposal distribution would take a long time with large sample set to get the samples from high probability region in the target distribution. Secondly, in high dimensional system, even if we obtain the samples in the high probability region of the target distribution, their importance weights also vary by large factors. It means that the approximation of the target distribution just depends on a few samples. These two are the root of the weight degeneracy problem in the typical Particle Filter [Fred *et al.*, 2011].

3.2.3 Markov Chain Monte Carlo

Another popular Monte Carlo method is Markov Chain Monte Carlo (MCMC). It is a sampling strategy where the samples generated from the proposal distribution construct a markov chain to explore the target state space. Due to the fact that this markov chain is irreducible and aperiodic, its stationary distribution exists and is equal to the target distribution. Here,

we briefly introduce two typical MCMC methods : Metropolis-Hastings algorithm and Gibbs Sampler [Walsh , 2004].

Metropolis-Hastings (MH) algorithm is one of the most practical MCMC methods, which draws a candidate sample x^* from proposal distribution $q(x^*|x)$ and accepts it with the probability $\alpha(x^*, x) = \min\{1, p(x^*)q(x|x^*)/(p(x)q(x^*|x))\}$:

1. Initialization : set i=0 and draw a start point $x^{(0)}$
2. For i=0,2,...N-1 :
   - sample $x^* \sim q(x^*|x^{(i)})$
   - sample $u \sim \mathcal{U}(0, 1)$
   - if $u < \alpha(x^*, x^{(i)}) = \min\{1, p(x^*)q(x^{(i)}|x^*)/(p(x^{(i)})q(x^*|x^{(i)}))\}$

     $$x^{(i+1)} = x^*$$

     else

     $$x^{(i+1)} = x^{(i)}$$

After setting the start point $x^{(0)}$, we sample a candidate point x^* from $q(x^*|x^{(i)})$ given the current point $x^{(i)}$, then the markov chain moves to x^* with acceptance rate $\alpha(x^*, x^{(i)})$. This step is illustrated by $u < \alpha(x^*, x^{(i)})$ in the algorithm where u is a random sample from $\mathcal{U}(0, 1)$. If $u < \alpha(x^*, x^{(i)})$, it means that $\alpha(x^*, x^{(i)})$ is large enough to accept the candidate.

Generally an independent proposal distribution $q(x^*|x^{(i)}) = q(x^*)$ is chosen to simplify the Metropolis-Hastings algorithm. However, the choice of proposal distribution is crucial for the sampling performance.

For example, the target distribution is a student's t distribution $p(x) = 1/\pi(1 + x^2)$ and proposal distribution is $N(0, \sigma^2)$. Using Metropolis-Hastings algorithm with 10000 samples, we got different approximation results with different proposal distribution. In figure 3.8 and 3.9, $\sigma = 1$ makes the samples well-mixed so that the approximated distribution is close to the true distribution.

On the contrary, $\sigma = 5$ in figure 3.10 and 3.11 leads the samples poor-mixed, where the approximated distribution is totally different from the target distribution.

Additionally, in order to make the samples well-mixed, generally the samples would be drawn to estimate the distribution after a burn-in period. This operation could ensure the stationary has been reached using MCMC.

The Gibbs Sampler is another MCMC algorithm. In fact, it is a special Metropolis-Hastings algorithm with the acceptance rate $\alpha = 1$ [Walsh , 2004]. The idea is to sequentially sample n variables from n univariate conditional distributions, and then the joint approximation distribution of n variables' samples simulates the full joint distribution. Here is the specific algorithm :

- Initialization : $x_{1:n}^{(0)}$
- For $i = 0, 2, ...N - 1$:
  - sample $x_1^{(i+1)} \sim p(x_1|x_2^{(i)}, x_3^{(i)}, \cdots, x_n^{(i)})$
  - sample $x_2^{(i+1)} \sim p(x_1|x_1^{(i+1)}, x_3^{(i)}, \cdots, x_n^{(i)})$

    $\vdots$

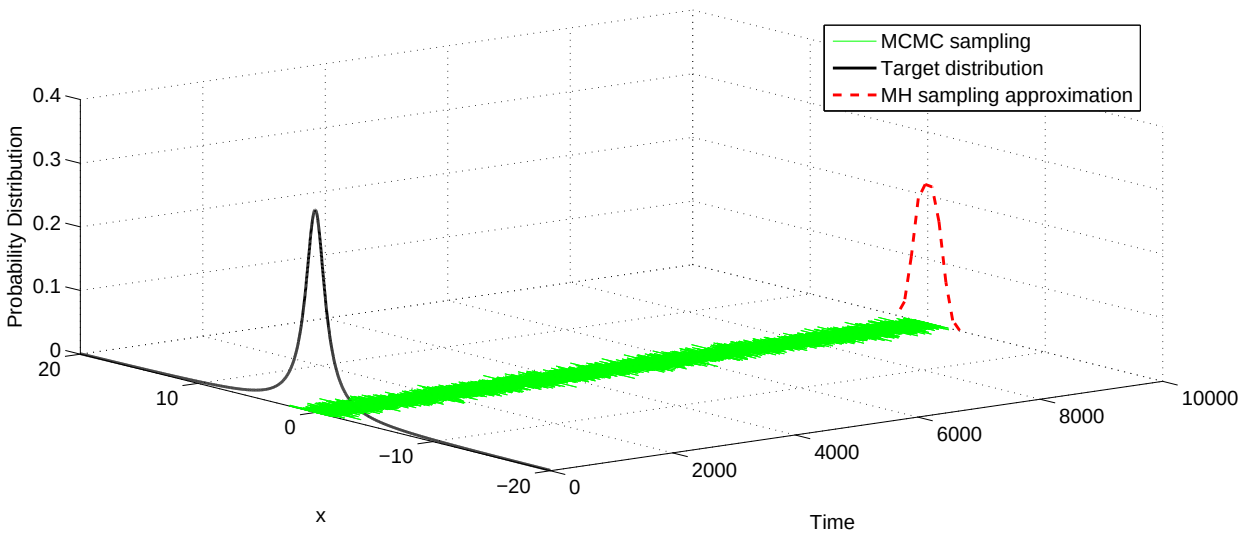

FIGURE 3.8: MH Sampling with $q(x^*) \sim N(0,1)$

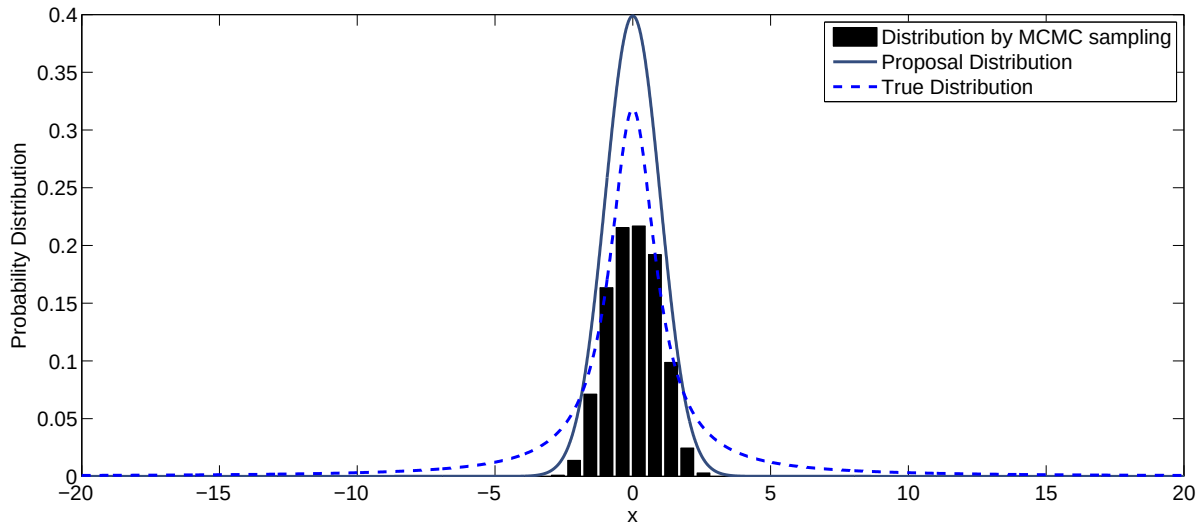

FIGURE 3.9: Estimated distribution by MH Sampling with $q(x^*) \sim N(0,1)$

– sample $x_n^{(i+1)} \sim p(x_n|x_1^{(i+1)}, x_2^{(i+1)}, \cdots, x_{n-1}^{(i+1)})$

Considering the following example in which we draw samples from a bivariate gaussian distribution $\mathbf{N}(\mu, \mathbf{P})$ using Gibbs Sampler with 2000 samples and assuming that

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.71}$$

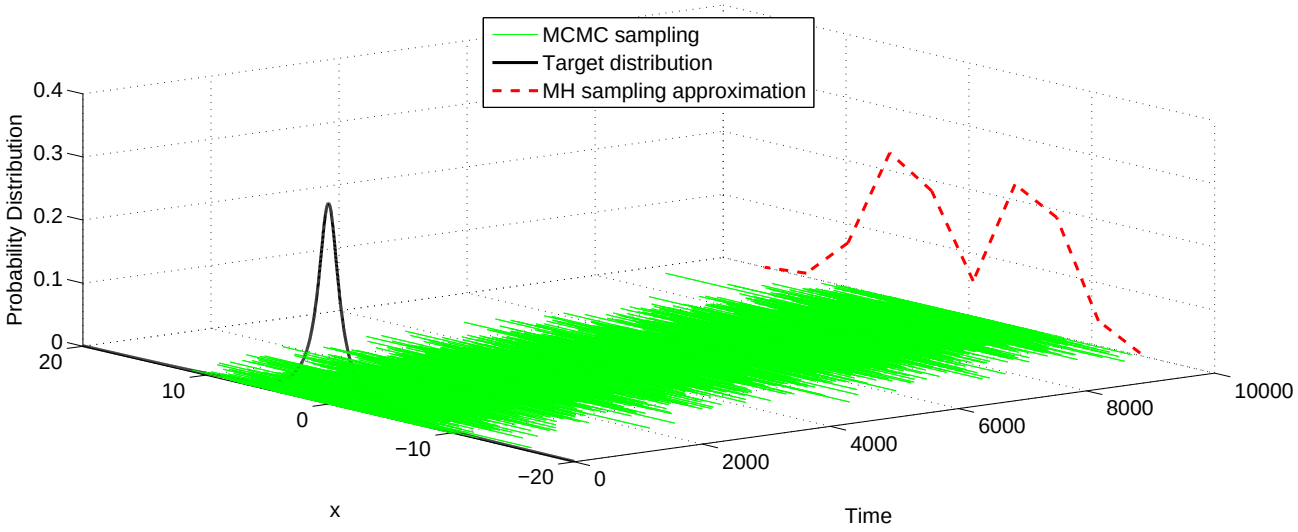

FIGURE 3.10: MH Sampling with $q(x^*) \sim N(0, 5^2)$

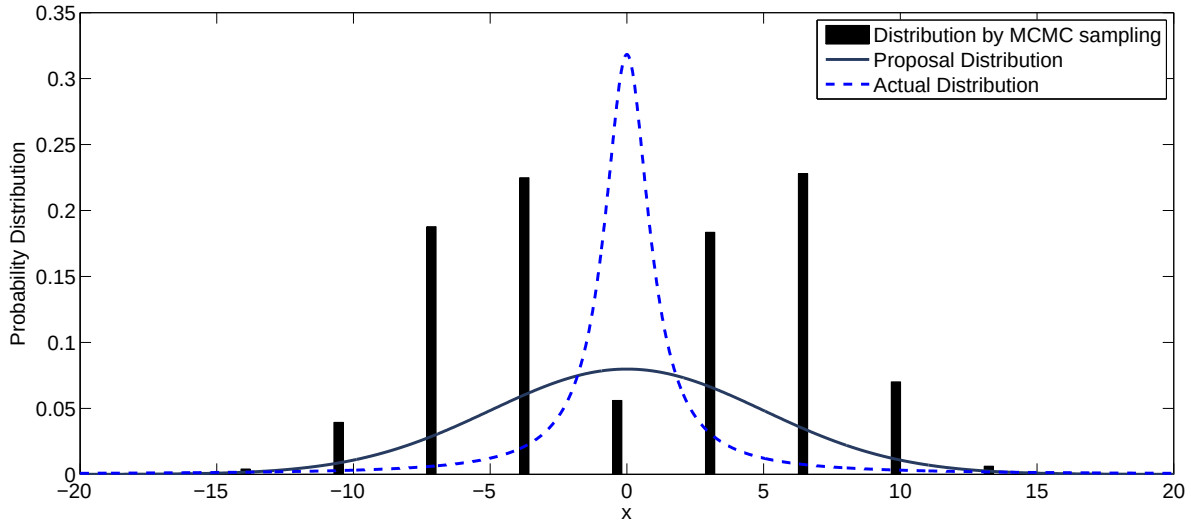

FIGURE 3.11: Estimated distribution by MH Sampling with $q(x^*) \sim N(0, 5^2)$

$$\mathbf{P} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \tag{3.72}$$

Then, the conditional distribution of x_1 given x_2 is $N(0.5x_2, 1 - (0.5)^2 = 0.75)$, similarly, the conditional distribution of x_2 given x_1 is $N(0.5x_1, 0.75)$. The figure 3.12 shows the path of Gibbs Sampling in the first 5 iterations, and the generated 2000 samples are shown in figure 3.13. Both the sampling results in figure 3.14 and 3.15 are shown that the estimated distributions

well approximate the true distributions.

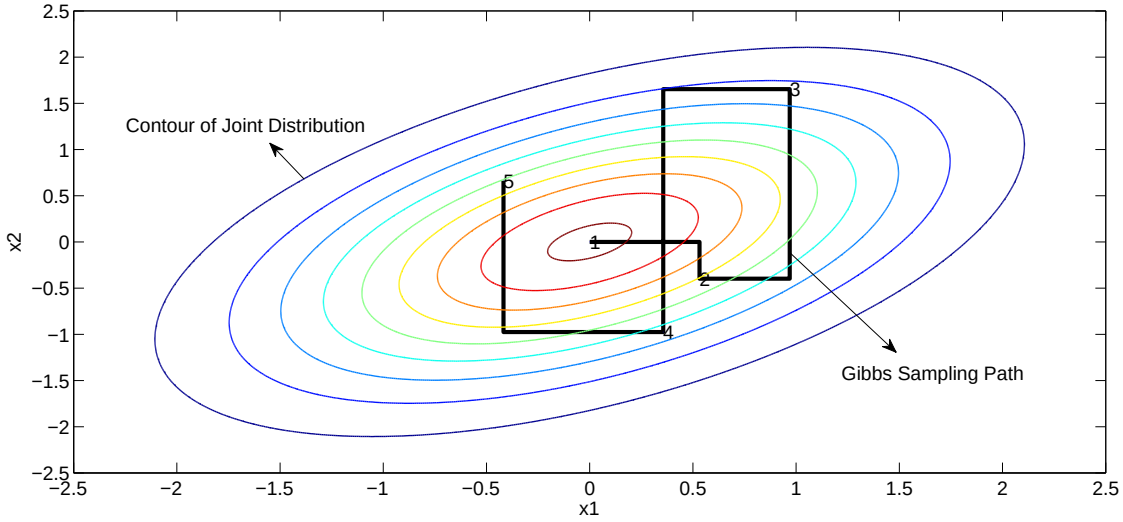

FIGURE 3.12: Path traversed by Gibbs Sampler in first 5 iterations

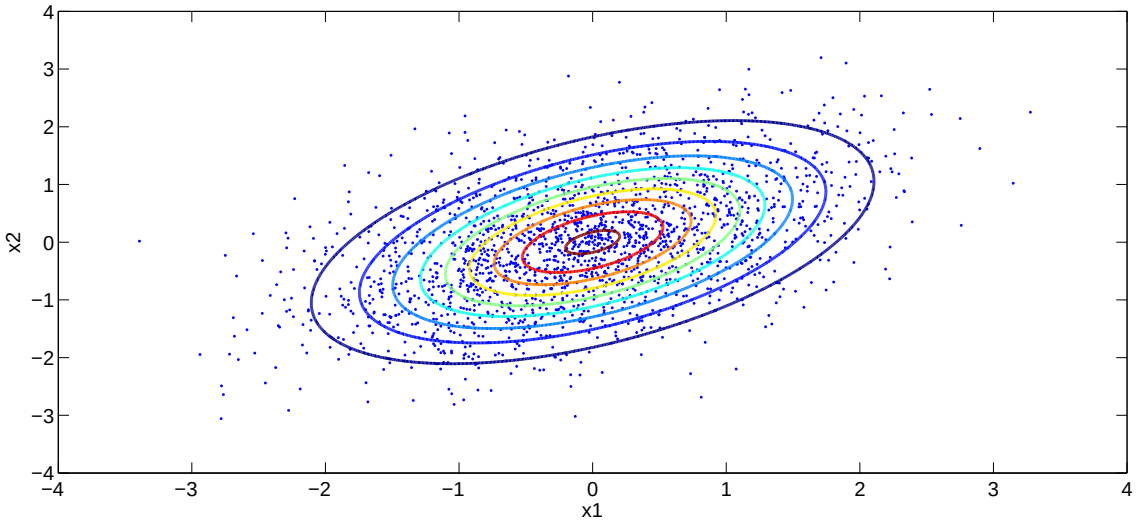

FIGURE 3.13: The 2000 samples generated by Gibbs Sampler

MCMC is a general framework for drawing the samples from complicated high-dimensional distributions. Metropolis-Hastings algorithm is simple and universal but the choice of proposal distribution is very crucial because the statistical properties of markov chain heavily depend on it [Christophe *et al.*, 2008]. A popular variant of Metropolis-Hastings algorithm is Gibbs Sampler which sequentially samples n variables from n univariate conditional distributions.

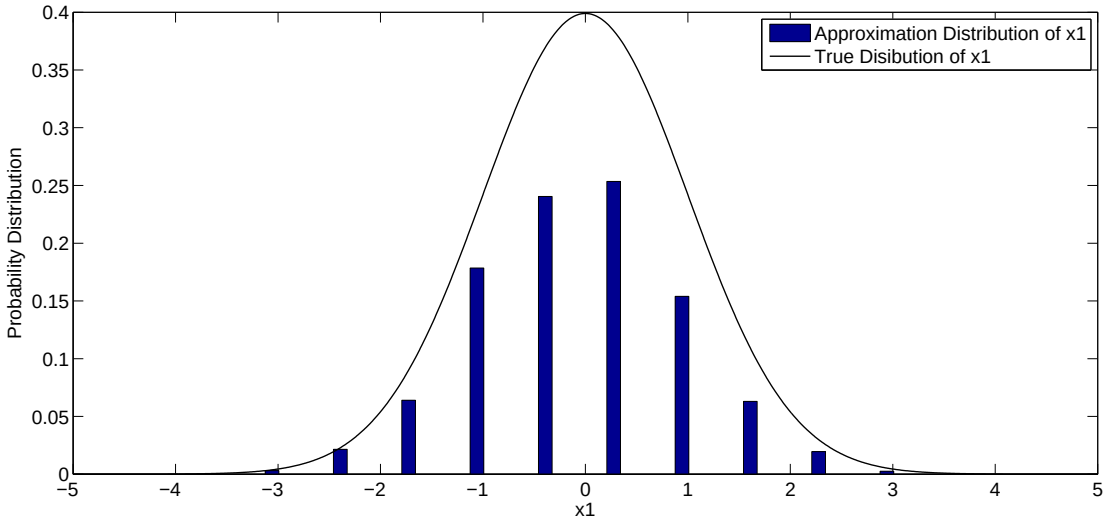

FIGURE 3.14: Estimated distribution of x_1 by Gibbs Sampler

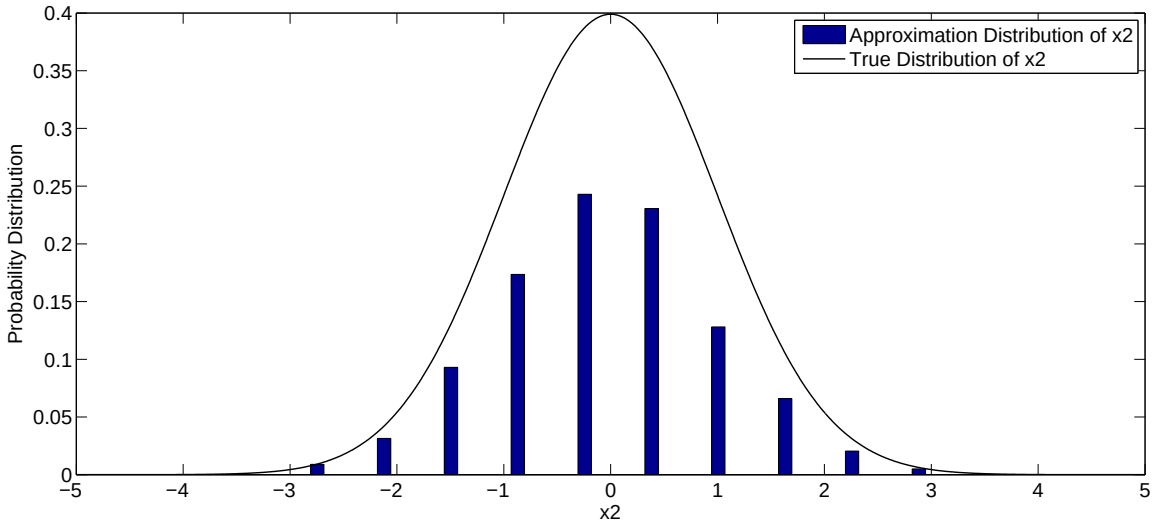

FIGURE 3.15: Estimated distribution of x_2 by Gibbs Sampler

However, it is still inefficient when highly dependent variables are not generated simultaneously [Andrieu *et al.*, 2008].

After introducing Monte Carlo methods, the question is how to apply them in to filtering problem. I would explain it with Particle Filter and Particle MCMC in the following sections.

3.3 Sequential Monte Carlo Estimation : Particle Filter

For the past decades, Monte Carlo methods have been considered as more robust algorithms to handle the nongaussian issue using stochastic sampling framework. For a general filtering problem $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$ which is an arbitrary posterior distribution of state sequence given observation sequence, if the samples $\mathbf{x}_{0:t}^{(i)}$ $(i = 1, \cdots, N)$ could be generated from $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$, Monte Carlo approximations are as follows :

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:t}) \tag{3.73}$$

$$E[g(\mathbf{x}_{0:t})] = \int g(\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})d\mathbf{x}_{0:t} \approx \hat{E}[g(\mathbf{x}_{0:t})] = \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}_{0:t}^{(i)}) \tag{3.74}$$

However, it's almost impossible to directly apply Monte Carlo Sampling methods for sequential state approximations. Hence, Monte Carlo approximation is applied iteratively to estimate the subtask at each time step, which is known as Sequential Monte Carlo Sampling procedure[Jun *et al.*, 1998; Arnaud *et al.*, 2000; Olivier *et al.*, 2007]. Using Sequential Monte Carlo techniques, the true nongaussian posterior distribution could be approximated by a class of Particle Filter.

One of the widely used Particle Filters is Sequential Importance Sampling Particle filter [Zhe, 2003], which is developed by approximating the important region of posterior distribution recursively. However, there exists one serious drawback called weight degeneracy problem in this Particle Filter - few weights will be nonzero after several iterations, especially in high dimensional system. It means that a great number of unimportant particles have to be calculated all the time, which leads to the inefficient performance [Zhe, 2003; Caglar *et al.*, 2011; Fred *et al.*, 2011; Samarjit, 2010]. In order to reduce weight degeneracy problem to improve approximation results, a Resampling step has been developed and used after Importance Sampling at each time step. This produces a well-known Sampling Importance Resampling Particle Filter [Arnaud *et al.*, 2000; Deok-Jin, 2005]. But this kind of Particle Filter just alleviates the degeneracy problem rather than solving it because Resampling step introduces high correlation between particles after a few iterations through replicating the particles with high important weight at each time step. The weight degeneracy is actually transformed into particle diversity problem, which is one of the underlying reasons why this Particle Filter framework has unsatisfactory estimation in high dimensional systems [Fred *et al.*, 2011].

In order to compensate the disadvantages of Sampling Importance Resampling Particle Filter, a number of its variants have been proposed recently. The first one is to choose a better proposal distribution which could approximate the true posterior distribution as close as possible. Extended Kalman Particle Filter and Unscented Particle Filter use a gaussian distribution as the proposal distribution by local linearization [Rudolph *et al.*, 2000]. It could make use of the current observation information to match the posterior density better. Secondly, by inserting Markov Chain Monte Carlo into the resampling step, MCMC particle filter would restore the particle diversity [Walter *et al.*, 2001; Francois *et al.*, 2008]. The third improvement is related to dynamic model analysis. Through dividing the system into linear and nonlinear parts, Rao-Blackwellized Particle Filter applies Kalman Filter to deal with the linear part and Particle Filter to approximate the nonlinear part in order to improve the computational

efficiency [Thomas *et al.*, 2005; Robert *et al.*, 2007]. Additionally, a KLD-Sampling Particle filter [Dieter, 2001] is designed to adjust the number of particles efficiently according to Kullback-Leibler divergence to ensure the approximated accuracy. Last but not least, Another direction is to combine Particle Filter into the MCMC framework to construct a so-called Particle MCMC Andrieu *et al.* [2010], which applies the approximated distribution by Particle Filter as the proposal distribution of MCMC. Even though the computational complexity is relatively high, Particle MCMC efficiently estimates the complicated posterior distribution in high dimensional space by the trivial prior proposal distribution in Particle Filter. Additionally, a novel particle learning method [Carlos *et al.*, 2010] is developed for filtering, sequential parameter learning and smoothing in a class of general dynamic systems. The performance is better than Particle Filter and MCMC.

The following subsections would introduce the general Particle Filter framework that typically consists of Sequential Importance Sampling and Resampling step to illustrate how it works.

3.3.1 Sequential Importance Sampling (SIS)

The idea of SIS is to apply importance sampling to draw N samples with N weights from a factor of $q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$ at each time step [Niclas, 1999; Jun *et al.*, 1998].

Let's reconsider the expectation problem using importance sampling view :

$$\begin{aligned} E[g(\mathbf{x}_{0:t})] &= \int g(\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})d\mathbf{x}_{0:t} \\ &= \int g(\mathbf{x}_{0:t})\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})}q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})d\mathbf{x}_{0:t} \end{aligned} \tag{3.75}$$

and define the weight :

$$\mathbf{w}_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{x}_{0:t}, \mathbf{y}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})} \tag{3.76}$$

Then, the expectation problem is transformed into a faction which is only related to the weight and the proposal distribution

$$E[g(\mathbf{x}_{0:t})] = \frac{\int g(\mathbf{x}_{0:t})\mathbf{w}_t(\mathbf{x}_{0:t})q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})d\mathbf{x}_{0:t}}{\int \mathbf{w}_t(\mathbf{x}_{0:t})q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})d\mathbf{x}_{0:t}} \tag{3.77}$$

Suppose now the proposal distribution could be expressed as the following factorized form :

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})q(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1}) \tag{3.78}$$

N samples $\mathbf{x}_t^{(i)}$ at time t could be drawn from $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$, meanwhile, the weight could be calculated recursively :

$$\begin{aligned} \mathbf{w}_t(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{x}_{0:t}, \mathbf{y}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})q(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})}\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \end{aligned} \tag{3.79}$$

According to the analysis above, the Monte Carlo approximations of state sequence are computed :

$$\hat{E}[g(\mathbf{x}_{0:t})] = \frac{\frac{1}{N}\sum_{i=1}^{N}\mathbf{w}_t(\mathbf{x}_{0:t}^{(i)})g(\mathbf{x}_{0:t}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N}\mathbf{w}_t(\mathbf{x}_{0:t}^{(i)})} = \sum_{i=1}^{N}\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})g(\mathbf{x}_{0:t}^{(i)}) \tag{3.80}$$

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \sum_{i=1}^{N}\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})\delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}) \tag{3.81}$$

The SIS framework is as follows :

- For $t = 0, 1, 2, ...T$
  - For $i = 1, 2, ...N$: sample $x_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ and set $x_{0:t}^{(i)} = \{x_{0:t-1}^{(i)}, x_t^{(i)}\}$
  - For $i = 1, 2, ...N$: calculate the weights by equation (3.79) and then normalize to $\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})$

Here we use the Univariate Nonstationary Growth Model in [Jouni *et al.*, 2007; Olivier *et al.*, 2007] to illustrate the weight degeneracy problem :

$$x_t = 0.5x_{t-1} + 25x_{t-1}(1 + x_{t-1}^2)^{-1} + 8\cos(1.2t) + w_{t-1} \tag{3.82}$$

$$y_t = 0.05x_t^2 + v_t \tag{3.83}$$

where the system and observation noise are $w_t \sim N(0, 10)$, $v_t \sim N(0, 1)$ respectively, the initial state distribution is $x_0 \sim N(0, 10)$. Additionally, we use $N = 50$ particles at each time step, time interval is 0.1 and the terminal time is 5.

By SIS methods, the state estimation result is shown in figure 3.16. The hidden state estimation by SIS is better than Extended Kalman Filter and Unscented Kalman Filter in figure 3.6 when the observation is relatively inaccurate. But SIS is not good enough because there exists the particle degeneracy problem in figure 3.17. It shows that only around 10 particles out of 50 have nonzero weights after 50 iterations, which leads to the useless computation.

Even though SIS makes the estimated process simpler with sequentially sampling, it exists a very serious problem which is known as weight degeneracy. The variance of the importance weights increases over time so that only few weights would be nonzero after several iterations [Arnaud *et al.*, 2000], which means that lots of computation waste on updating non-contributed weights.

3.3.2 Resampling Step

In order to solve weight degeneracy problem, the intuitive idea is to reduce particles with small weights and focus on particles with large weights. This procedure could be done in resampling step. There are many popular resampling methods [Zhe, 2003; Arnaud *et al.*, 2008], here we briefly introduce Multinomial Sampling and Systematic Resampling algorithm.

The basic idea of Multinomial Sampling is to draw N samples from the current particles according to their normalized weights so that the important particles with large weight could be propagated to improve the performance.

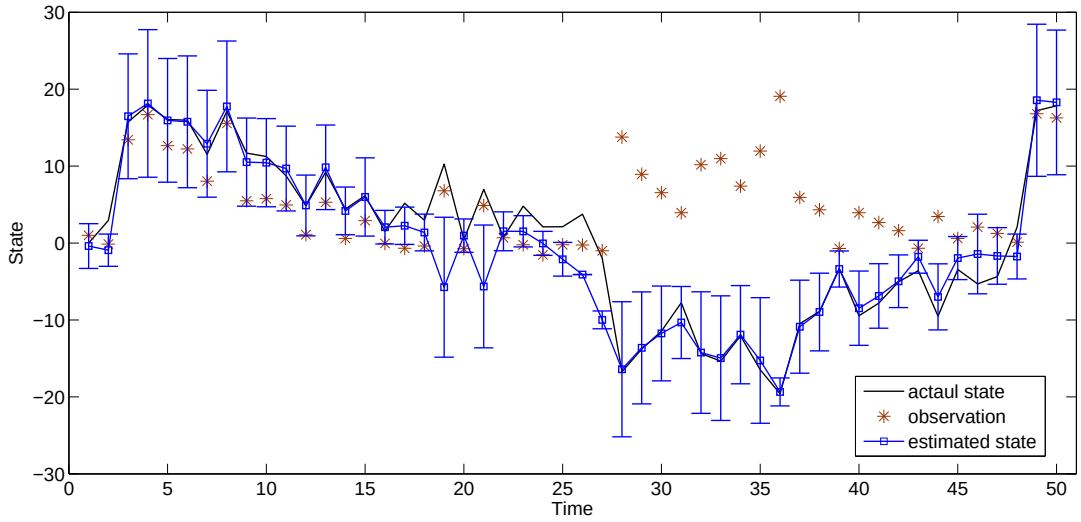

FIGURE 3.16: State Estimation by SIS over time

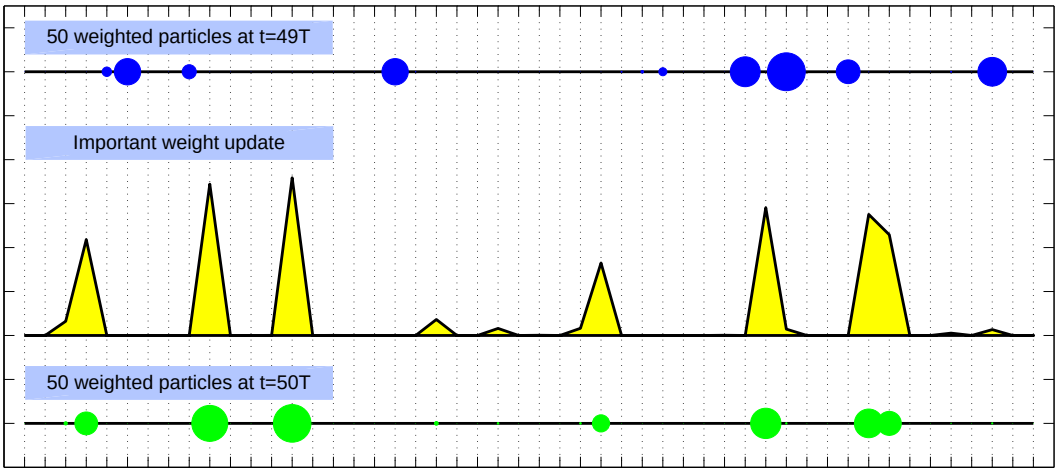

FIGURE 3.17: SIS operation from 49T to 50T

Another popular resampling method is the following Systematic Resampling which considers the weights as continuous randomly ordered variables in the interval (0, 1), and it is a minimum variance sampling algorithm [Zhe, 2003].

1. Set $i = 1$ and $c_1 = 0$
2. For $i = 2, ...N$: $c_i = c_{i-1} + \tilde{w}_t^{(i)}$
3. Set $k = 1$

4. Sample $u_1 \sim \mathcal{U}[0, 1/N]$
5. For $j = 1, 2, ...N$:
   - $u_j = u_1 + (j-1)/N$
   - While $u_j > c_k$, $k = k + 1$
   - Otherwise, $x_t^{(j)} = x_t^{(k)}$ and $\tilde{w}_t^{(i)} = 1/N$

Additionally, from the particle efficiency perspective, a measure of particle degeneracy could be used as an effective sample size [Zhe, 2003; Deok-Jin, 2005; Haug, 2005]

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N_p} (\tilde{w}_t^{(i)})^2} \tag{3.84}$$

then if N_{eff} is smaller than the predefined threshold $N_T = 0.5N_p$ or $2N_p/3$, resampling step is performed using resampling methods, otherwise, we neglect this procedure and go to the next time instant.

3.3.3 Sampling Importance Resampling (SIR) Particle Filter

Sampling Importance Resampling Particle Filter (SIRPF) is a typical stochastic sampling method for recursive state estimation problem, which combines Importance Sampling and Resampling at each time step to deal with the weight degeneracy problem in SIS.

Generally, a suitable proposal distribution is crucial for approximation performance. The optimal proposal distribution which minimizes the variance on the importance weights is given by [Arnaud *et al.*, 2000; Deok-Jin, 2005; Arnaud *et al.*, 2008]

$$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) \tag{3.85}$$

However, it is infeasible to sample from this proposal distribution. Hence, in order to simplify SIRPF framework, the prior distribution of state sequence is chosen as the proposal distribution

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0) \prod_{k=1}^{t} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{3.86}$$

which means

$$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.87}$$

Therefore, N samples $\mathbf{x}_t^{(i)}$ at time t could be drawn from $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the weight could be updated recursively :

$$\begin{aligned} \mathbf{w}_t(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})} \mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \\ &= p(\mathbf{y}_t|\mathbf{x}_t)\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \end{aligned} \tag{3.88}$$

The generic SIR Particle Filter framework is as follows [Pierre *et al.*, 2006; Gordon *et al.*, 1993; Michael *et al.*, 2009; Caglar *et al.*, 2011] :

- For $n = 0, 1, 2, ...T$
  1. For $i = 1, 2, ...N$: sample $x_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and set $x_{0:t}^{(i)} = \{x_{0:t-1}^{(i)}, x_n^{(i)}\}$

2. For $i = 1, 2, ...N$: calculate the weights according to equation (3.88) and then normalize to $\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})$
3. resample the current N particles according to their weights to obtain N new particles with equal weights $1/N$

Considering the same Univariate Nonstationary Growth Model in [Jouni *et al.*, 2007; Olivier *et al.*, 2007], all the conditions are the same except that a resampling step is added after Importance step at each time. In figure 3.18. The estimation is closer to the true hidden state than SIS method and the degeneracy problem is done with resampling step, which is shown in figure 3.19.

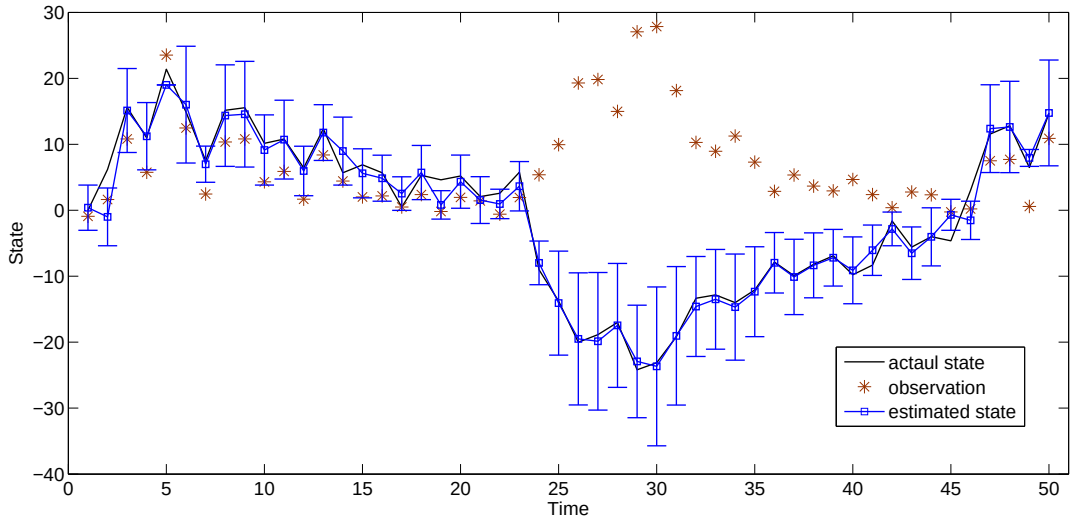

FIGURE 3.18: State Estimation by SIR over time

3.3.4 Discussion on SIR Particle Filters

SIR Particle Filter has been widely used in the real nonlinear and nongaussian systems, for example, it could be used for target tracking in wireless senor network [Huiying *et al.*, 2010], mobile robot navigation and localization [Niclas, 1999] and computer vision [Samarjit, 2010].

However, there are serval drawbacks [Caglar *et al.*, 2011; Fred *et al.*, 2011; Samarjit, 2010; Jinxia *et al.*, 2010; Peter *et al.*, 2008]. Firstly, the resampling step effectively alleviates particle degeneracy problem, but meanwhile it introduces the sample diversity impoverishment because of copying the important particles with large weights. Secondly, SIR Particle Filter lies in the simplicity due to apply the transition probability as the proposal distribution at each time step, however, the new information from the observation $\mathbf{y}_t$ is not contained in this proposal distribution, which makes the filter sensitive for the outliers and predict blindly, especially when the likelihood is relatively narrow. Thirdly, Particle Filter requires relatively high computation that is time exponential in n. n is the dimension of the state vector. Hence the effective number

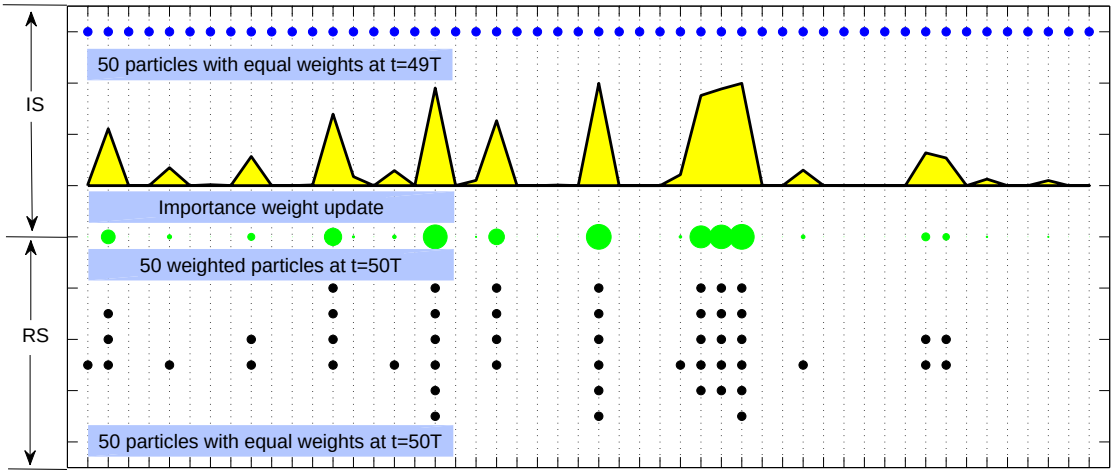

FIGURE 3.19: SIR operation from 49T to 50T, where IS represents Importance Sampling and RS represents Resampling.

of particles is a very crucial factor for computational efficiency, especially in high dimensional systems.

3.4 Particle Markov Chain Monte Carlo

Sequential Monte Carlo and Markov Chain Monte Carlo are the most two popular sampling methods for state estimation, however, both of them have unsatisfactory performance in the complicated high dimensional dynamic systems. SIR Particle filter reduces weight degeneracy problem via resampling, but this operation will introduce sample diversity impoverishment. Metropolis-Hastings algorithm is one of the best known MCMC algorithm that draws the samples from high-dimensional distributions, but the produced markov chain heavily depends on whether the proposal distribution is good or not [Christophe *et al.*, 2008].

In this context, the recent Particle Markov Chain Monte Carlo method [Andrieu *et al.*, 2010, 2008] provides a general framework to incorporate Sequential Monte Carlo into MCMC, which takes advantage of the merits in both methods in order to update the estimation performance further in high dimensional cases.

3.4.1 Particle Metropolis-Hastings Sampler

Particle MCMC follows the idea which considers the approximation distribution given by Particle Filter as the proposal distribution in MCMC. It aims to approximate the high-dimensional complicated posterior distribution using the trivial low dimensional proposal distribution by Particle Filter.

A standard Metropolis-Hastings algorithm for filtering is to sample a candidate $\mathbf{x}^{\star}_{0:t}$ from a proposal distribution $q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$ and accept it with the rate

$$\alpha = min\{1, \frac{p(\mathbf{x}^{\star}_{0:t}|\mathbf{y}_{0:t})q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})}{p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})q(\mathbf{x}^{\star}_{0:t}|\mathbf{y}_{0:t})}\} \tag{3.89}$$

Generally, it's not feasible to get the optimal proposal $q^{opt}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$. Hence, it is natural to consider a good approximation of true posterior as a proposal distribution in MCMC. Particle filter algorithm provides a convenient way for sampling using MCMC since its approximation distribution is represented by discrete weighted particles.

By derivation from [Andrieu *et al.*, 2010], the general Particle Metropolis-Hastings algorithm is as follows :

1. $i = 0$, run a Particle Filter algorithm and calculate the approximated marginal distribution $\hat{p}(\mathbf{y}_{0:t})(0)$, then draw a sample $\mathbf{x}_{0:t}(0) \sim \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$
2. For iteration $i \geq 1$
   - run a Particle Filter and calculate the approximated marginal distribution $\hat{p}(\mathbf{y}_{0:t})^{\star}$, then draw a sample $\mathbf{x}^{\star}_{0:t} \sim \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$
   - with rate

$$\alpha = min\{1, \frac{\hat{p}(\mathbf{y}_{0:t})^{\star}}{\hat{p}(\mathbf{y}_{0:t})(i-1)}\} \tag{3.90}$$

     accept $\mathbf{x}_{0:t}(i) = \mathbf{x}^{\star}_{0:t}$ and $\hat{p}(\mathbf{y}_{0:t})(i) = \hat{p}(\mathbf{y}_{0:t})^{\star}$
   - otherwise

$$\mathbf{x}_{0:t}(i) = \mathbf{x}_{0:t}(i-1) \tag{3.91}$$

$$\hat{p}(\mathbf{y}_{0:t})(i) = \hat{p}(\mathbf{y}_{0:t})(i-1) \tag{3.92}$$

This algorithm could be applied to approximate complicated distributions with high precision while only requiring the low dimensional proposal design in Particle Filter.

Considering the nonlinear system in [Andrieu *et al.*, 2010] as an example :

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x^2_{t-1}} + 8\cos(1.2t) + w_t \tag{3.93}$$

$$y_t = \frac{x^2_n}{20} + v_t \tag{3.94}$$

where the system and observation noise are assumed as gaussian distribution $w_t \sim N(0, 20)$, $v_t \sim N(0, 20)$ respectively, the initial state distribution is $x_0 \sim N(0, 5)$. Time interval is set to 0.1 and the terminal time is 5, then the objective is specified as $p(\mathbf{x}_{0:50}|\mathbf{y}_{0:50})$. Additionally, the iteration of MCMC is 1000.

In figure 3.20, the estimated state sequence using Particle MCMC correctly approximates the true latent state sequence with relatively small variance even though the observation has relatively large noise at average level. Figure 3.21 shows that the approximated posterior distribution is well-fitted for true hidden state distribution at $t = 50$.

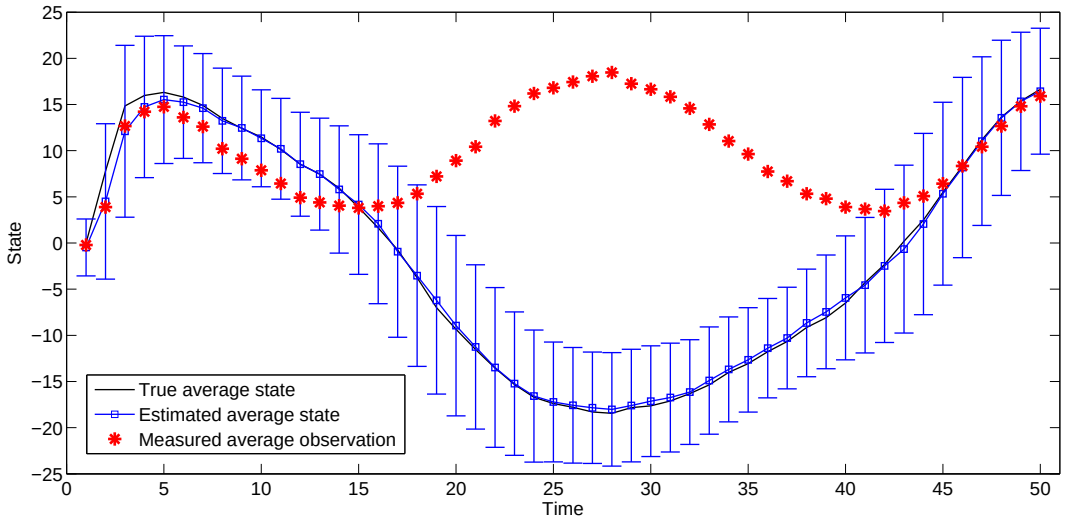

FIGURE 3.20: State estimation over time given observation

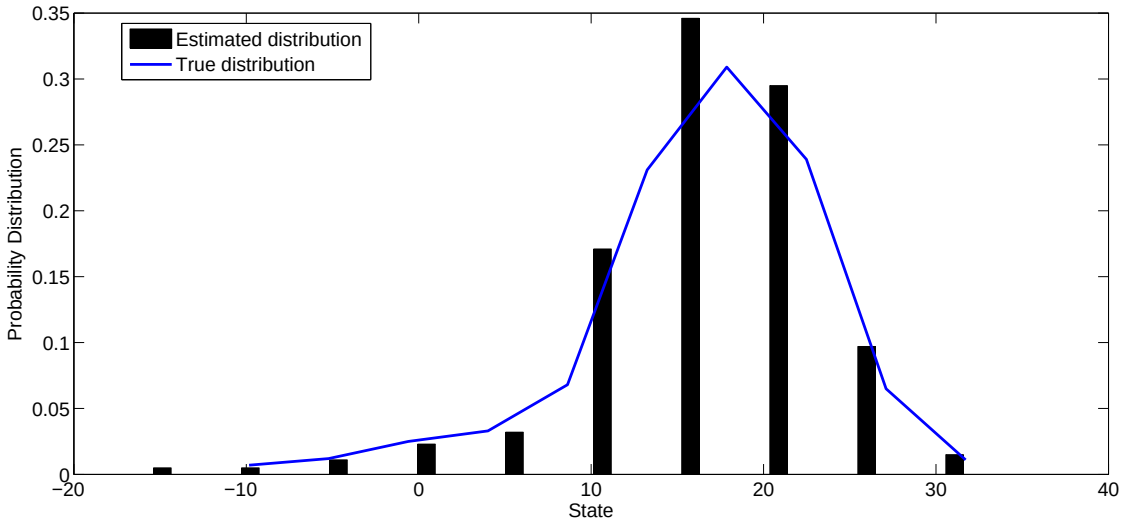

FIGURE 3.21: Posterior distribution approximation at $t = 50$

3.4.2 Discussion on Particle MCMC

Particle MCMC applies the estimated posterior distribution by Particle filter as a proposal distribution in MCMC. It could be more robust since it is less likely to suffer from the weight degeneracy problem which is one of the best known drawbacks in Particle Filter [Andrieu *et al.*, 2010]. In fact, Particle MCMC does not need Particle Filter to provide an accurate posterior approximation due to the fact that the estimated density by particle filter just returns a sample

in MCMC.

Lately, some extensions of Particle MCMC have been developed to improve the performance further for general dynamic systems. [Gareth*et al.*, 2010] incorporates an adaptive MCMC into Particle MCMC framework so that the new method could adaptively learn the important region of marginal distribution for unknown model parameters in a general nonlinear dynamic system. [Chopin*et al.*, 2011] applies Particle MCMC as the MCMC rejuvenation steps in the iterated batch importance sampling (IBIS) to automatically tune the parameters with the same complexity as Particle MCMC under some certain condition. [Whiteley*et al.*, 2009] designs a new Particle MCMC for a class of multiple change-point problems. However, there still exists some improvements in Particle MCMC [Andrieu *et al.*, 2010]. Firstly, other more efficient Particle Filters or Smoothers would be a better proposal in MCMC. Then, a parallel chain or Population-Based MCMC algorithms could be used to increase the sample diversity.

According to this Chapter, there exists some drawbacks in the current methods for estimation problem. Hence, I propose to apply nonparametric techniques to solve these problems, and then apply the proposed methods in the real dynamic systems.

4

Research Objectives and Approach

After pointing out the drawbacks of the existing Bayesian Filtering algorithms, I should now explain what is my ultimate objective. My objective aims to apply nonparametric techniques to construct a general online estimation framework to overcome the shortcomings of current methods so that it could be applied in the real constrained environments with high accuracy as well as computational efficiency.

4.1 Research Objectives

The specific goals are as follow :

1. Improving the estimation performance of Particle MCMC
   a) Applying an efficient Particle Filter or Smoother to MCMC
   b) Implementing MCMCs in parallel to increase the sample diversity
2. Designing an efficient proposal for Particle Filter using nonparametric methods
   a) Using nonparametric methods to learn the optimal sequential proposal distribution
   b) Solving weight degeneracy problem by nonparametric methods
3. Designing a filtering framework for nonparametric dynamic system
   a) Designing a filtering framework for a dynamic model based on Gaussian Process
   b) Solving the estimation problem for a dynamic model based on Dirichlet Process
4. Applying the proposed nonparametric filtering methods into smoothing algorithms
5. Incorporating the designed nonparametric filter framework into nonlinear control
6. Implementing/validating our algorithms on toy examples and then on realistic applications

4.2 Methodologies

4.2.1 Methodologies for objective 1

Particle MCMC incorporates the approximated distribution by Particle Filter as the proposal distribution in MCMC to form a general framework to estimate the high dimensional systems. Hence, there are two sub-perspectives to improve the performance of Particle MCMC : one is to design an efficient Particle Filter or Smoother for MCMC, and the other is to implement MCMCs in parallel to increase the sample diversity.

4.2.1.1 Methodologies for subobjective 1a)

One of the main reasons why the recent Particle MCMC has high computation is that SIR particle filter fixes the number of particle. This would lead computational waste, especially when the true posterior distribution changes vastly over time. For example, in the beginning of global localization, mobile robot is with high uncertainty. We need a large number of particles to approximate the posterior distribution to get the relatively accurate position. However, once the robot knows where it is, we actually just need a very small number of particles to obtain its position. Hence, if the number of particle is fixed, the computation is less efficient. KLD-Sampling Particle Filter is a good way to adjust the particle number according to Kullback-Leibler divergence to improve the approximation quality [Dieter, 2001]. Hence, i plan to incorporate KLD-Sampling Particle Filter into Particle MCMC method to improve the accuracy and efficiency of Particle MCMC.

The proposed methodology to tackle the first sub-objective is to use KLD-Sampling Particle Filter instead of SIR Particle Filter as a proposal distribution in MCMC to improve the computational efficiency.

4.2.1.2 Methodologies for subobjective 1b)

In the basic Particle MCMC, Particle Filter would be operated at each iteration in MCMC. It is a kind of computational waste since all but only one particle from Particle Filter would be abandoned at each MCMC iteration [Andrieu *et al.*, 2010].

The possible methodology is to apply a parallel MCMC or Population-Based MCMC into Particle MCMC framework to take advantage of all the particles so that the estimation performance would be better because of increasing the particle diversity.

4.2.2 Methodologies for objective 2

The proposal distribution in Particle Filter is one of the most crucial factors to determine whether its estimation is good or not. The following methodologies aim to deal with the two subobjectives : one is to use nonparametric methods to learn the optimal sequential proposal distribution , the other is to solve weight degeneracy problem by nonparametric methods.

4.2.2.1 Methodologies for subobjective 2a)

The optimal proposal in the weight update fashion is $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$. However, it is impossible to get it because it is actually unknown. For simplicity, the typical SIR Particle Filter applies the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as a proposal distribution. But it does not consider the current observation information $\mathbf{y}_t$. In other words, we blindly draw the samples from $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ without adding the observation information, which would heavily affect estimation result.

For example, the weight update of SIR Particle Filter in practice is

$$\begin{aligned}\mathbf{w}_t(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)}\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \\ &= p(\mathbf{y}_t|\mathbf{x}_t)\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \end{aligned} \tag{4.1}$$

Since we do resampling at each time step, the weights at $t-1$ are the same which means that $\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1})$ could be deleted. Then

$$\mathbf{w}_t(\mathbf{x}_{0:t}) = p(\mathbf{y}_t|\mathbf{x}_t) \tag{4.2}$$

In figure 4.1, the high probability region of the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the low probability region of target distribution $p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Hence the samples we blindly draw from proposal distribution are not located at the important region of target distribution, which leads the fact that their weight distribution according to likelihood distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ is not possible to approximate the target distribution. From the likelihood distribu-

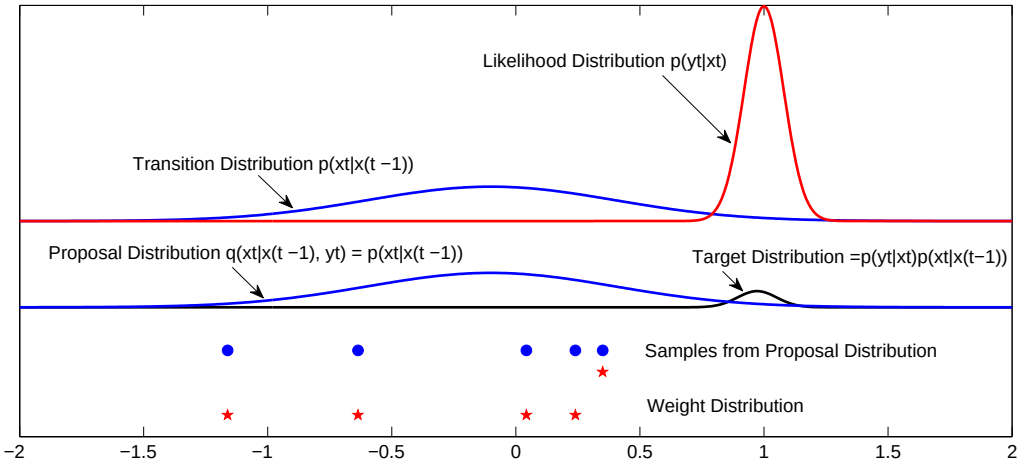

FIGURE 4.1: Important Sampling at one time step in Particle Filter

tion, we could see the variance is relatively small which means that the current observation accurately describes the current state. But proposal distribution does not consider it. This is the underlying reason for poor estimation performance.

Hence, in order to add the current observation information to improve the approximation, the methodology for the first subgoal is to use nonparametric methods to learn the optimal

proposal distribution. In [Jonathan *et al.*, 2009; Jonathan , 2011; Marc *et al.*, 2009], the gaussian process is used to learn the state space model for filtering problem. In fact, they apply gaussian process to learn the transition and likelihood distribution. According to what they did, the proposed methodology expects that a gaussian process could learn the proposal distribution. Since the optimal proposal distribution at one time actually reflects the nonlinear relationship between $[\mathbf{x}_{t-1}, \mathbf{y}_t]$ and $\mathbf{x}_t$, learning this distribution is actually a general gaussian process regression problem. Furthermore, if we treat this as a regression problem, we could also consider other regression methods in which the prediction is a probability form .

4.2.2.2 Methodologies for subobjective 2b)

Even though the subobjective 2a) would improve the estimation performance by adding current observation information, the modified Particle Filter still have the weight degeneracy problem. The resampling step hides this problem by replicating the particles with large weight, but it will bring the particle diversity loss problem. Additionally, resampling step will also lead time-consuming problem.

Hence, the motivation is how to solve the weight degeneracy problem without resampling step. Looking back to the original importance sampling procedure, we find

$$\mathbf{w}_t(\mathbf{x}_t) \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_t|\mathbf{y}_{1:t})} \tag{4.3}$$

If the weight does not update in a sequential fashion and just use the above form, the weight degeneracy problem could be done [Jan, 2011]. Hence the methodology for the second subobjective is to approximate the total optimal proposal $q(\mathbf{x}_t|\mathbf{y}_{1:t})$ and prediction distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ by gaussian process regression or other nonparametric methods.

4.2.3 Methodologies for objective 3

Generally a filtering problem assumes that the dynamic state space model has been already known, however, it is not the case in the real world. Hence the goal is to develop a general filtering framework for the unknown dynamic model. The first thing is to learn the unknown dynamic model from the training data set. Since the traditional parametric models use a finite number of parameters, it would suffer from over-fitting or under-fitting of data when the model complexity and the amount of data do not match. Hence, more and more attention has been focused on nonparametric model. The representative two are gaussian process and dirichlet process. Both of them could be applied to establish the Hidden Markov Model, which allows us to incorporate filtering techniques to implement the general estimation.

4.2.3.1 Methodologies for subobjective 3a)

[Jonathan *et al.*, 2009; Jonathan , 2011] uses gaussian process to learn the state space model and then applies it with different kinds of filtering algorithm. The limitation is that they use the ground true state in the training set, which may not be possible in the real applications. Therefore, they extended their gaussian process Bayesian filter with modified Gaussian Processes Latent Variable Model [Jonathan *et al.*, 2009, 2011; Lawrence, 2005] which

is original from Gaussian Processes Dynamical Model[Jack*et al.*, 2008]. Hence the methodology is to learn optimal proposal for Particle Filter using Gaussian Processes Dynamical Model.

4.2.3.2 Methodologies for subobjective 3b)

The Dirichlet process is a stochastic process which could be applied as a bayesian nonparametric model for training data, especially in the Dirichlet process mixture models. [Caron *et al.*, 2008] uses a Dirichlet process mixture to model the unknown noise distribution in the linear and gaussian dynamic system, and then develops an off-line MCMC and an online particle filter for estimation problem. Additionally, the hierarchical Dirichlet process hidden Markov model (HDP-HMM) and its variants [Beal *et al.*, 2002; Yee *et al.*, 2010; Emily, 2009; Emily *et al.*, 2007] have been established for the past decades, which provide us a good way to design the general filtering framework with bayesian nonparametric models. [Emily *et al.*, 2007] uses HDP-HMM in a linear gaussian dynamic model with unknown control input, then Kalman filter is applied for estimation problem. However, filtering for both models mentioned before is just for linear and gaussian system. Hence the methodology is to apply Dirichlet process mixture or HDP-HMM into the general nonlinear dynamic model, and then combine different Monte Carlo methods (MCMC, Sequential Monte Carlo) to solve the estimation problem.

4.2.4 Methodologies for objective 4

As mentioned before, if the state space model is unknown, we should learn it from the training set. In fact, learning requires more smoothing rather than filtering because smoothing provides better estimation of the state in the process [Stuart *et al.*, 2010; Carlos *et al.*, 2010]. Hence the idea is to apply the previous proposed nonparametric filtering framework into smoothing algorithms for a general dynamic system in order to improve the approximation efficiency.

4.2.5 Methodologies for objective 5

Control task is actually the final goal for a general dynamic system after learning the model and estimating the state [Nadine *et al.*, 2007]. The recent articles [Villiers *et al.*, 2011; Stahl *et al.*, 2011] successfully combine particle filter into a nonlinear model predictive controller design in order to increase the control robustness.

Hence the methodology is to use the designed nonparametric filter framework into nonlinear predict model control to improve the control performance. Additionally, filtering technique is actually a state estimation method, if we could use the estimated state as a feedback which could be used as input of nonlinear controllers design, the close loop control system could have good transitional and steady performance.

4.2.6 Methodologies for objective 6

Filtering technique could be applied in a great number of real applications, such as visual-based human motion tracking, financial analysis, weather broadcasting and so on. Hence, the methodology is to implement the algorithms on toy examples and then on realistic applications.

One of the most important real applications is autonomous robot localization and navigation, which would be considered as my main implementation plan.

4.2.7 Work Package

Here are three main work packages : formal theory studies, methods evaluation and complexity analysis, practical application implementation.

Formal theory studies : the existing filtering techniques unsatisfactorily performs when the posterior distribution of a dynamic system is complicated and nongaussian. The underlying reason is that there are some drawbacks in their theoretical design frameworks. Hence, it is necessary to study the formal theory in depth, which is the foundation of my research work.

1. Further study on the Filtering and smoothing Theory to broaden the design framework for a general dynamic system :
   - Further study on Particle MCMC and KLD-Sampling Particle Filter to find the combination manner : [Stuart *et al.*, 2010; Zhe, 2003; Arnaud *et al.*, 2000; Olivier *et al.*, 2007; Deok-Jin, 2005; Arnaud *et al.*, 2008; Haug, 2005; Andrieu *et al.*, 2010, 2008; Dieter, 2001; Thrun *et al.*, 2005; Carlos *et al.*, 2010; Gareth*et al.*, 2010; Chopin*et al.*, 2011; Whiteley*et al.*, 2009]
   - Review the mathematical derivation of existing filtering and their own drawbacks : [Kalman, 1960; Maria, 2004; Daniel *et al.*, 1972; Zhe, 2003; Simon *et al.*, 2004; Jun *et al.*, 1998; Arnaud *et al.*, 2000; Olivier *et al.*, 2007; Fred *et al.*, 2011; Jinxia *et al.*, 2010; Peter *et al.*, 2008; Carlos *et al.*, 2010].
   - Finish the study of other novel filtering methods [Choo *et al.*, 2001; Guo *et al.*, 2006; Andreasen , 2008; Friston, 2008]. This will provide many other creative frameworks which could be used in the proposed methods since the design generally needs the combination of different filtering algorithms.
   - Kernel Adaptive Filtering : [Weifeng *et al.*, 2010]. It aims to get a general idea how the nonparametric kernel methods could be applied in the filtering problem.
   - Smoothing techniques : [Stuart *et al.*, 2010; Heijden *et al.*, 2004; Carlos *et al.*, 2010].
2. Further study on the Nonparametric Theory systemically in order to incorporate suitable techniques for filtering in complex dynamic phenomenon :
   - Firstly, further study on Gaussian Process [Christopher , 2006; Carl *et al.*, 2006] so that i could fully understand its merits and shortcomings. Secondly, further study on how the gaussian process could be used to establish nonparametric state space model [Jonathan *et al.*, 2009; Jonathan , 2011; Jonathan *et al.*, 2011; Lawrence, 2005; Jack*et al.*, 2008; Hartikainen *et al.*, 2010; Marc *et al.*, 2009], especially when we just know the observation sequence that is the common case in the real world.
   - Understand Dirichlet distribution, Dirichlet Process, Dirichlet Process Mixture, Hierarchical Dirichlet Process and how to apply it into hidden Markov model [Bela *et al.*, 2010; Caron *et al.*, 2008; Beal *et al.*, 2002; Yee *et al.*, 2010; Emily, 2009; Emily *et al.*, 2007].
   - Other nonparametric methods for opening my mind : [Stuart *et al.*, 2010; Zhe, 2003].
3. Apply the state estimated by the nonparametric filter to develop the novel nonlinear controller to make the close loop control system more robust : [Nadine *et al.*, 2007; Villiers *et al.*, 2011; Stahl *et al.*, 2011; Tapani*et al.*, 2009].

Methods evaluation and complexity analysis : after sufficiently studying all the theory, I will design the nonparametric filtering/smoothing framework for the general dynamic systems, and then analyze the computational complexity and prove its validation. Theorem proof and performance analysis of proposed methodologies are very important since this step could show whether the designed methods are valid or not at theoretical level, whether the computational efficiency is good enough for the practical applications in the real world.

Practical application implementation :

A critical reason why we need to design a novel filter is that the estimated results should ensure the estimated accuracy while alleviating the computational cost. Here are serval work packages for the implementation issue with data from real system.

1. Master the operations of real robot platform
2. Programme the nonparametric filter framework to realize the given tasks
3. Evaluate the performance by error statistics and comparison with current methods

According to what i plan to do, i will introduce the research work i'm currently doing in the following chapter.

5

Current Work and Preliminary Results

Currently, there are two specific parts of my work whose general goal is to make more accurate approximation of Bayesian filtering :

1. The first work is to combine a KLD-Sampling Particle Filter into Particle MCMC framework so that the designed algorithm could adjust the particle quantity and quality in the Particle Filter to obtain a better proposal distribution in MCMC to improve the estimated performance.
2. The second work is to apply gaussian process regression to learn a proposal distribution, which incorporates the current observation information to obtain an approximation optimal proposal distribution, to improve the estimated performance for general nonlinear and nongaussian systems.

5.1 Adaptive Particle Markov Chain Monte Carlo

There are two widely used sampling methods for general high-dimensional probability distribution : Sequential Monte Carlo and Markov Chain Monte Carlo. However, the shortcomings in both algorithms will severely restrict their applications in practice. The well known weight degeneracy problem in SIR Particle Filter could be alleviated by resampling, but the particle diversity will be reduced over time. The MCMC methods is theoretically feasible, but the practical performance would deteriorate when the proposal distribution is not well chosen. Intuitively, a lot of attention has been paid on whether these two methods could be efficiently combined together to form a general framework. Particle Markov Chain Monte Carlo method has been developed lately by applying the approximation posterior by Sequential Monte Carlo as a proposal in MCMC in order to update the estimation performance further in high dimensional cases [Andrieu *et al.*, 2010, 2008].

However, the traditional SIR Particle Filter used in Particle MCMC fixes particle number, which causes the estimation to have high computational complexity and low approximation

efficiency, especially when true posterior changes vastly over time. A well-known variant named KLD-Sampling Particle Filter could deal with this problem adaptively using Kullback-Leibler divergence.

Hence, the idea is to apply KLD-Sampling Particle Filter into Particle MCMC Framework to ensure the approximate performance with high efficiency.

5.1.1 KLD-Sampling Particle filter

It's known that the number of the particles greatly affects the inference performance. In the traditional particle filter, this number is always fixed, which leads to poor performance because the complexity of posterior distribution changes vastly over time [Dieter, 2001]. In order to improve estimation quality, KLD-Sampling particle filter is proposed [Dieter, 2001; Thrun *et al.*, 2005] to adjust the number of particles adaptively over time via bounding the inference error with Kullback-Leibler divergence (KLD) :

$$KL[\hat{p}(x)\|p(x)] = \sum_x \hat{p}(x) \log \frac{\hat{p}(x)}{p(x)} \tag{5.1}$$

where $\hat{p}(x)$ is the discrete approximated distribution of $p(x)$.

For any discrete distribution $p(x)$ with k different bins, the number of samples drawn from $\hat{p}(x)$ is defined :

$$N = \frac{\chi^2_{k-1,1-\delta}}{2\varepsilon} \tag{5.2}$$

so that KLD between the true discrete distribution and its maximum likelihood estimation is smaller than ε with confidence $1-\delta$. In the formula, χ^2_{k-1} is the chi-square distribution with $k-1$ degrees of freedom.

For the state estimation problem, the true posterior distribution of state sequence is unknown, which means that the number of bins k is unknown. Hence at each time step in the particle filter, k is incremental if there is a new sampled particle which does not belong to the previous bins. The KLD particle filter algorithm is shown as follows :

1. Initialize ε and δ
2. For $t = 1, 2, ...T$, set $N = 0$, $k = 0$
   a) Sample a particle $x_{t-1}^{(i)}$ with the normalized weights at $t-1$
   b) Sample a particle $x_t^{(N)} \sim p(x_t|x_{t-1}^{(i)})$
   c) Calculate its weight according to $p(y_t|x_t^{(N)})$
   d) if ($x_t^{(N)}$ falls into an empty bin b) then
      i. $k = k + 1$
      ii. Set b non-empty
   e) $N = N + 1$
   f) if $N > \frac{1}{2\varepsilon}\chi^2_{k-1,1-\delta}$, return $x_t^{(i)}$ $(i = 1, 2, ...N)$. Otherwise, return to a)

At each time step, KLD-Sampling particle filter calculates the necessary particle number to estimate the true posterior distribution in the conventional particle filter framework, then the Monte Carlo approximation could be represented with time-variant particle number.

5.1.2 KLD-Sampling Particle Markov Chain Monte Carlo

Particle MCMC is a simple way to estimate state sequence problem with high accuracy, however, its relatively expensive tradeoff is computational efficiency because it actually belongs to MCMC algorithms, which takes a long time to converge. Here, we consider this problem by replacing the standard Particle Filter with KLD-Sampling particle filter in the previous Particle MCMC Framework. By doing so, the proposed KLD-Sampling Particle MCMC reduces the computation through improving the quantity and quality of the particles adaptively.

1. $i = 0$, run KLD-Sampling particle filter and calculate the approximated marginal distribution $\hat{p}(\mathbf{y}_{0:t})(0)$, then draw a sample $\mathbf{x}_{0:t}(0) \sim \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$
2. For iteration $i \geq 1$
   - run KLD-Sampling particle filter and calculate the approximated marginal distribution $\hat{p}(\mathbf{y}_{0:t})^{\star}$, then draw a sample $\mathbf{x}^{\star}_{0:t} \sim \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$
   - with rate

$$\alpha = min\{1, \frac{\hat{p}(\mathbf{y}_{0:t})^{\star}}{\hat{p}(\mathbf{y}_{0:t})(i-1)}\} \tag{5.3}$$

   accept $\mathbf{x}_{0:t}(i) = \mathbf{x}^{\star}_{0:t}$ and $\hat{p}(\mathbf{y}_{0:t})(i) = \hat{p}(\mathbf{y}_{0:t})^{\star}$
   - otherwise

$$\mathbf{x}_{0:t}(i) = \mathbf{x}_{0:t}(i-1) \tag{5.4}$$

$$\hat{p}(\mathbf{y}_{0:t})(i) = \hat{p}(\mathbf{y}_{0:t})(i-1) \tag{5.5}$$

Due to the fact that KLD-Sampling particle filter is modified by the traditional particle filter using Kullback-Leibler divergence, which could be applied in conjunction with any scheme [Dieter, 2001], the validity of KLD-Sampling Particle MCMC could be proved with the same theorem in [Andrieu *et al.*, 2010].

5.1.3 Numerical Illustration

Considering the Univariate Nonstationary Growth Model :

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + w_t \tag{5.6}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{5.7}$$

where the system and observation noise are assumed as gaussian distribution $w_t \sim N(0, 20)$, $v_t \sim N(0, 20)$ respectively, the initial state distribution is $x_0 \sim N(0, 5)$.

Time interval is set to 0.1 and the terminal time is 5, then the objective is specified as $p(\mathbf{x}_{0:50}|\mathbf{y}_{0:50})$. In KLD-Sampling particle filter, both ε and δ are set to 0.05. Additionally, the iteration of MCMC is 1000.

The simulations are shown as follows by running the KLD-Sampling Particle MCMC and Particle MCMC respectively. In figure 5.1, the estimated state sequence using KLD-Sampling particle MCMC correctly approximates the true latent state sequence with relatively small variance even though the observation has relatively large noise at average level.

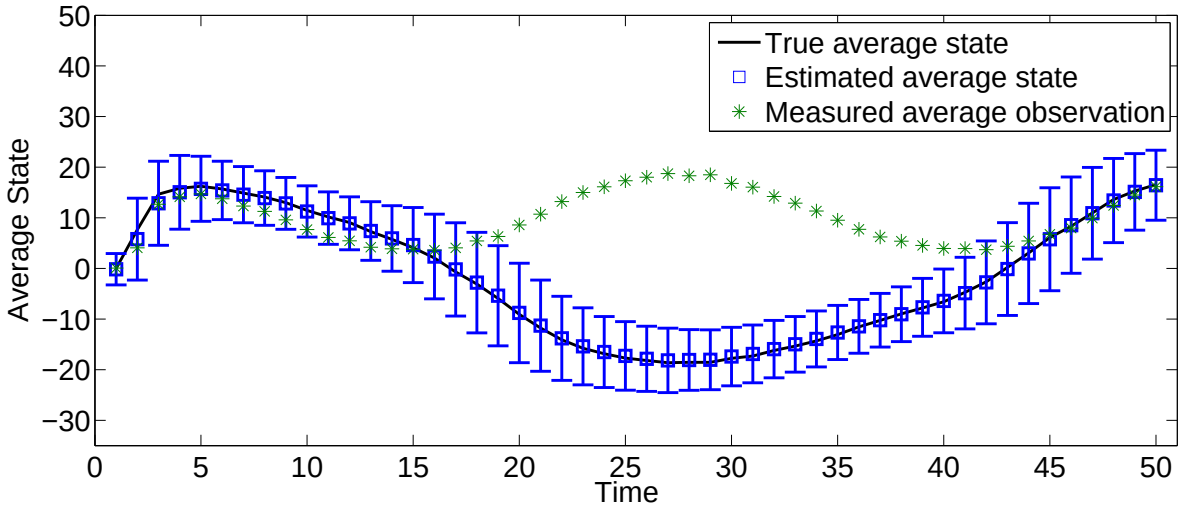

FIGURE 5.1: State estimation over time given observation

In 5.2, the average number of particles in the proposed algorithm is smaller than 100 over time.

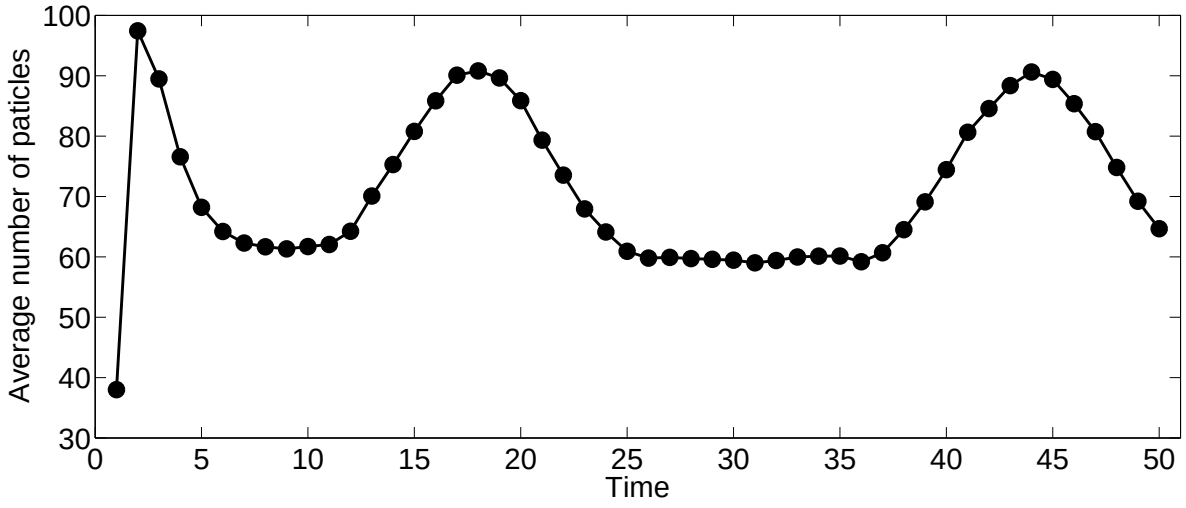

FIGURE 5.2: Average number of particles over time

We then run standard Particle MCMC in which the particle number is fixed as $N = 100$. Using root mean squared error ($RMSE$) as the evaluation criteria

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2} \tag{5.8}$$

we got the following result :

	PMCMC	KLD-Sampling PMCMC
$RMSE$	0.7033	0.4874

TABLE 5.1: $RMSE$ Comparison

As we could see, our KLD-Sampling Particle MCMC performs better than standard Particle MCMC.

Additionally, the acceptance rate over iteration and MCMC samples drawn from the proposed algorithm at $t = 50$ are respectively shown in figure 5.3 and 5.4. Both results show that the samples are well mixed.

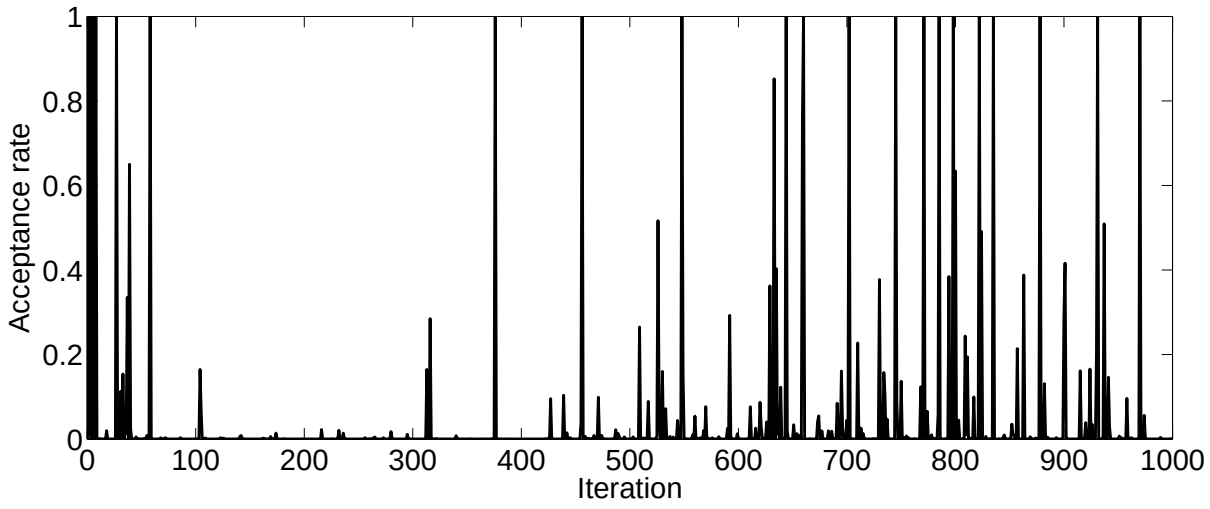

FIGURE 5.3: Acceptance rate over iteration

Finally, figure 5.5 shows that the approximated posterior distribution is well-fitted for true hidden state distribution at $t = 50$. This also validates our KLD-Sampling Particle MCMC.

Above all, the simulation clearly represents that KLD-Sampling PMCMC applies smaller size of particles and simpler proposal distribution to get better approximation and higher efficiency.

I have submitted this work to Signal Processing Letters [Wang*et al.*, 2011].

5.2 Gaussian Process Based Particle Filter

One of the most successful techniques for state estimation is Particle Filter. The traditional SIR Particle filter is based on Importance Sampling and Resampling at each time step, hence the proposal distribution is a very crucial factor determining whether the estimated performance is acceptable.

The optimal proposal distribution is given by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) \tag{5.9}$$

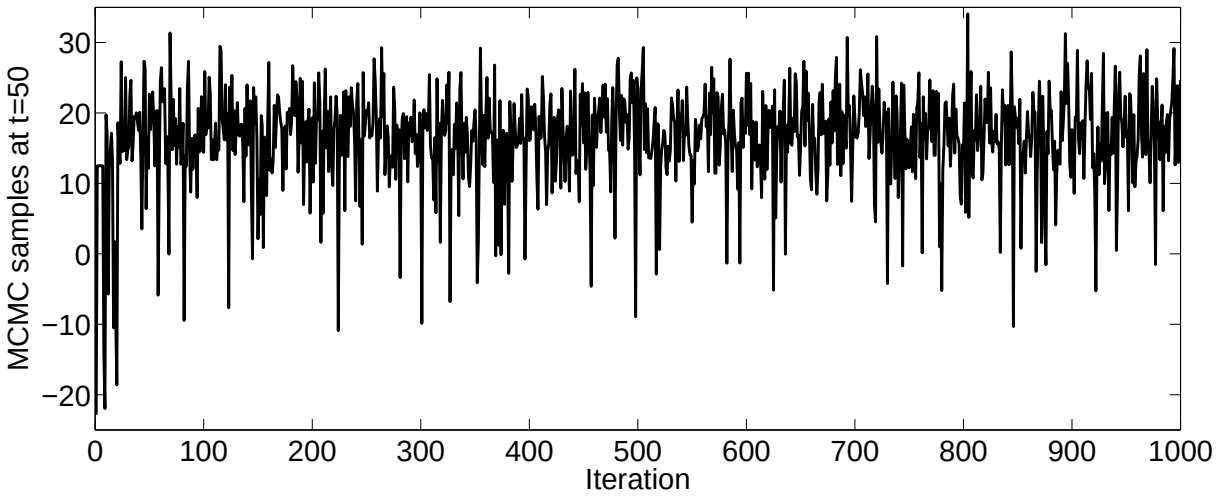

FIGURE 5.4: Samples at $t = 50$ by KLD-Sampling Particle MCMC

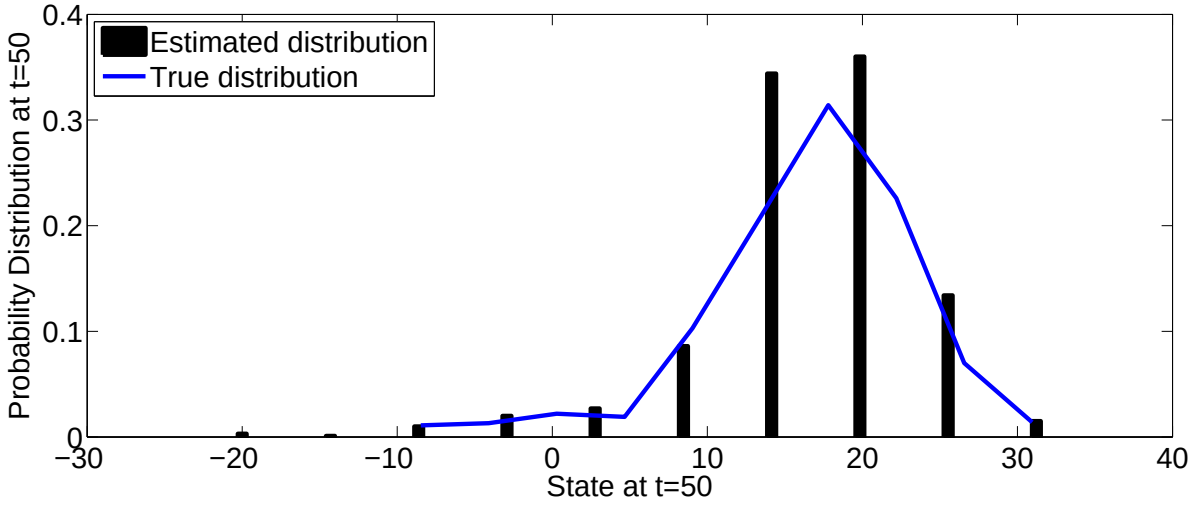

FIGURE 5.5: Posterior distribution approximation at $t = 50$

However, it is difficult to sample $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ because it's unknown. Therefore, SIR Particle Filter applies the parametric transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as the proposal distribution. This simple consideration neglects the important information of the current observation $\mathbf{y}_t$, which leads to the poor estimation.

Therefore, the idea is to use nonparametric gaussian process to learn the optimal proposal distribution to improve the approximated performance and alleviate the weight degeneracy problem.

5.2.1 Gaussian Process Regression

A gaussian process is a collection of random variables , any finite number of which have a joint gaussian distribution [Carl *et al.*, 2006]. It is a nonparametric method which represents a gaussian distribution over functions.

Assume the training set with n data is $D = < X,\ \mathbf{y} >$, where $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ is the input matrix and $\mathbf{x}_i$ is d dimensional. $\mathbf{y} = [y_1, y_2, \cdots, y_n]$ is the output vector, a regression problem could be solved by the gaussian process as follows [Christopher , 2006; Jonathan *et al.*, 2009; Carl *et al.*, 2006] :

$$y_i = f(\mathbf{x}_i) + \epsilon \tag{5.10}$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$ and f is a nonlinear function. Based on the training set $D = < X,\ \mathbf{y} >$ and a test input $\mathbf{x}_*$, the gaussian process defines a gaussian predictive distribution over the target output y_* :

$$p(y_*|\mathbf{x}_*, D) = N(y_*; \mu(\mathbf{x}_*, D), \Sigma(\mathbf{x}_*, D)) \tag{5.11}$$

with the mean

$$\mu(\mathbf{x}_*, D) = \mathbf{k}_*^T [K + \sigma_\epsilon^2 I]^{-1} \mathbf{y} \tag{5.12}$$

and variance

$$\Sigma(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T [K + \sigma_\epsilon^2 I]^{-1} \mathbf{k}_* \tag{5.13}$$

where k is the kernel function of gaussian process. $\mathbf{k}_*$ is a vector that represents the kernel values between the test input and the training set, $\mathbf{k}_*[i] = k(\mathbf{x}_*, \mathbf{x}_i)$. K is the kernel matrix of the training set, $K[i, j] = k(\mathbf{x}_i, \mathbf{x}_j)$.

The most widely used kernel function is gaussian kernel :

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 exp[-0.5(\mathbf{x} - \mathbf{x}')W(\mathbf{x} - \mathbf{x}')^T] \tag{5.14}$$

where σ_f^2 is the signal variance related to nonlinear function f, W defines the smoothness of the process. Additionally, the hyperparameters σ_ϵ^2, σ_f^2 and W could be learned using numerial optimization methods such as conjugate gradient ascent [Carl *et al.*, 2006].

5.2.2 Particle Filter Based on Gaussian Process

The goal is to apply gaussian process to learn the optimal proposal distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$. In fact, this proposal distribution reflects the nonlinear model between $[\mathbf{x}_{t-1}, \mathbf{y}_t]$ and $\mathbf{x}_t$:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{y}_t) + \epsilon \tag{5.15}$$

where $\mathbf{g}$ is a nonlinear unknown function, $\epsilon \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$.

The problem now is actually transformed to use gaussian process to learn this model. The training set in this model becomes $D = < [\mathbf{x}_{t-1}^{training}, \mathbf{y}_t^{training}], \mathbf{x}_t^{training} >$ and then the gaussian process model is :

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) \approx \mathbf{N}(\mu([\mathbf{x}_{(t-1)}, \mathbf{y}_t], D), \Sigma([\mathbf{x}_{(t-1)}, \mathbf{y}_t], D)) \tag{5.16}$$

Then the resulting gaussian process based particle filter is :

– For $n = 0, 1, 2, ...T$

1. For $i = 1, 2, ...N$: sample $x_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) \approx \mathbf{N}(\mu([\mathbf{x}_{(t-1)}, \mathbf{y}_t], D), \Sigma([\mathbf{x}_{(t-1)}, \mathbf{y}_t], D))$ and set $x_{0:t}^{(i)} = \{x_{0:t-1}^{(i)}, x_n^{(i)}\}$
2. For $i = 1, 2, ...N$: calculate the weights according to

$$\mathbf{w}_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})} \tag{5.17}$$

   and then normalize to $\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})$
3. resample the current N particles according to their weights to obtain N new particles with equal weights $1/N$

5.2.3 Numerical Illustration

Considering the previous Univariate Nonstationary Growth Model :

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + w_t \tag{5.18}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{5.19}$$

where the system and observation noise are assumed as gaussian distribution $w_t \sim N(0, 10)$, $v_t \sim N(0, 1)$ respectively, the initial state distribution is $x_0 \sim N(0, 10)$. Time interval is set to 0.01 and the terminal time is 0.5, then the objective is specified as $p(\mathbf{x}_{0:50}|\mathbf{y}_{0:50})$. The number of particle is 10 at each time. With the simplicity, we set the hyperparameters $\sigma_\epsilon^2 = 1$, $\sigma_f^2 = 1$ and $W = I$. In the training set, we use 50 training data at each time step.

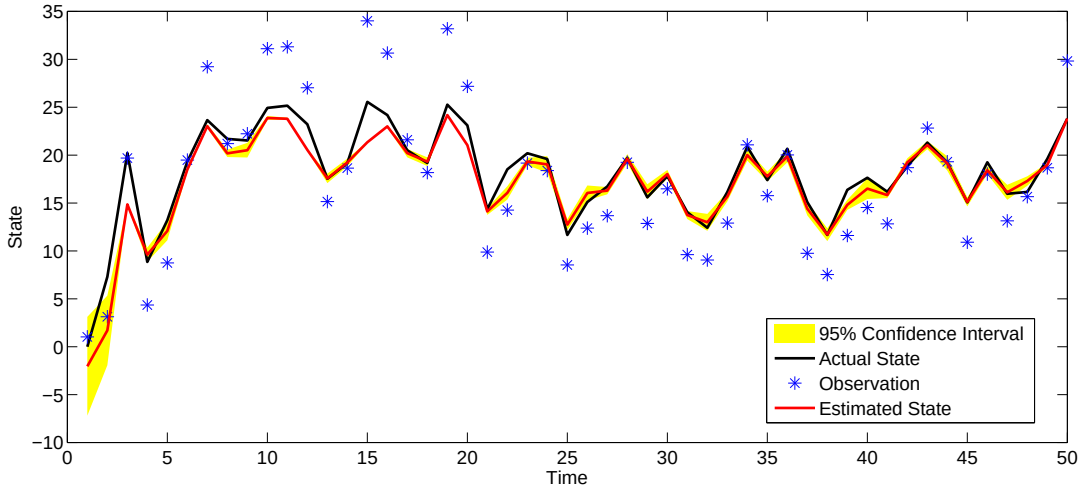

FIGURE 5.6: State Estimation by Gaussian Process Based Particle Filter

In the figure 5.6, the estimated state by Gaussian Process Based Particle Filter is well fitted with actual state. Then using root mean squared error ($RMSE$) as the evaluation criteria, we compare the performance of different filters :

	RMSE
Extened Kalman Filter	475.4430
SIR Particle Filter with 10 particles	2.0183
SIR Particle Filter with 100 particles	1.7209
Gaussian Process Based Particle Filter with 10 particles	1.5775

TABLE 5.2: $RMSE$ Comparison

From the results, we could see the estimated performance of Gaussian Process Based Particle Filter with 10 particles is the best among the Extended Kalman Filter and SIR Particle Filter. we also notice that Gaussian Process Based Particle Filter with 10 particles is even better than SIR Particle Filter with 100, which could improve the computational efficiency.

I'm submitting this work to ICRA 2012.

In order to make the proposed plan more implementable, I will provide a work plan to illustrate what I will do in the rest of my PhD studies.

6

Work Plan and Implementation

6.1 Work Plan

After introducing my research objective, methodology and current work, generally here is my future work plan with time schedule for the rest of my PhD :

1. Formal theory study
   a) Improving Particle MCMC algorithm : Fall 2011 to Spring 2012
   b) Mastering Bayesian nonparametric theory : Fall 2011 to Summer 2012
   c) Investigating Bayesian nonparametric methods for state estimation and control : Fall 2011 to Summer 2012
2. Elaborating the algorithms and analyzing performance
   a) Develop the bayesian nonparametric techniques for estimation and control problem : Spring 2012 to Fall 2013
   b) Theoretical simulation and performance analysis : Summer 2012 to Fall 2013
   c) Experimenting the proposed algorithm on real systems : Fall 2012 to Fall 2013
   d) Validation via article publication : Summer 2012 to Fall 2013
3. PhD thesis writing and defense : Spring 2014 to Summer 2014

6.2 Potential Obstacles

There are serval potential design issues along with the nonparametric filtering framework design and implementation :

1. Whether the proposal distribution designed by nonparametric methods could be sampled efficiently : the samples could be easily drawn from the proposal distribution which should be considered when choosing the nonparametric methods.

2. Whether the designed proposal distribution by nonparametric methods could be computational efficiency : if we plan to overcome the weight degeneracy problem of the typical particle filter, we should design the proposal distribution to approximate the true posterior distribution as close as possible. This could lead that the training data set should be large enough to get a good nonparametric density estimation.
3. The combination manner of nonparametric models and filtering framework : one of the theoretical problems is how to design a feasible Sequential Monte Carlo method for a complicated nonparametric dynamic system. In the real world, we generally do not know the form of dynamic model, hence nonparametric methods could be used to learn the underlying state space model. However, there exists a question about whether Sequential Monte Carlo or MCMC is suitable for these nonparametric model. In other words, the learned nonparametric model should be a form of probability distribution and not very complicated to implement. Hence, understanding both theories is the first step, then how to design the manner of cooperation is the most important issue.
4. Whether the design algorithm is too complicated for real system. The real robot system needs to estimate in a online manner in order to finish the given task, so we should keep in mind that it's necessary to design accurate nonparametric filtering with appropriate computational load.

7

Conclusion

The general objective for my research proposal is to use nonparametric techniques to deal with the general estimation problems. Through the combination of different sampling methods with adaptive scheme and a better proposal distribution design, we expect the result would improve the estimation accuracy. Additionally, in order to solve the real applications, we attempt to propose a filtering and smoothing framework for nonparametric bayesian model where we should learn the unknown dynamic system by nonparametric methods. Finally, all the techniques should be used to solve the nonlinear control problem in robotics, and we will apply error comparison and computational complexity analysis to evaluate the proposed algorithms' validity.

Bibliographie

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of basic Engineering*, 82(Series D)(1960), 35-45.

A. Jazwinski Stochastic processes and filtering theory, (1970).

Greg Welch and Gary Bishop. An Introduction to the Kalman Filter http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf, (2001).

N.A.Thacker and A.J.Lacey. The Likelihood Interpretation of the Kalman Filter. http://www.tina-vision.net/docs/memos/1996-002.pdf, (2006).

Tristan Fletcher. The Kalman Filter Explained. http://www.tristanfletcher.co.uk/LDS.pdf, (2010).

Nathan Funk. A Study of the Kalman Filter applied to Visual Tracking. http://www.njfunk.com/research/courses/652-probability-report.pdf, (2003).

Stuart J.Russell and Peter .Norvig Artificial Intelligence : A Modern Approach (Thrid Edition), (2010).

Shoudong Huang. Understanding Extended Kalman Filter - Part III : Extended Kalman Filter. http://services.eng.uts.edu.au/~sdhuang/, (2010).

Maria Isabel Ribeiro. Kalman and Extended Kalman Filters : Concept, Derivation and Properties. http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf, (2004).

Evgeni Kiriy and Martin Buehler. Three-state Extended Kalman Filter for Mobile Robot Localization. http://kom.aau.dk/group/05gr999/reference_material/filtering/eKf-3state.pdf, (2002).

Daniel L. Alspach and Harold W. Sorenson. Nonlinear Bayesian Estimation Using Gaussian Sum Approximations. *IEEE Transactions on Automatic Control*, 17 (1972), 439-448.

Zhe Chen. Bayesian Filtering : From Kalman Filters to Particle Filters, and Beyond. http://www.cse.buffalo.edu/~peter/refs/DataAssimilation/GenRefsTutorial%s/Chen_2003.pdf, (2003).

B. D. O. Anderson and J. B. Moore. Optimal Filtering, (1979).

Jayesh H. Kotecha and Petar M. Djuric. Gaussian Sum Particle Filtering. *IEEE Transactions on Signal Processing*, 51 (2003), 2602-2612.

Kazufumi Ito and Kaiqi Xiong. Gaussian Filters for Nonlinear Filtering Problems. *IEEE Transactions on Automatic Control*, 45 (2000), 910-927.

Simon J. Julier, Jeffrey K. Uhlmann and Hugh F. Durrant-Whyte. A New Approach for Filtering Nonlinear Systems. *Proceedings of the American Control Conference*, 3 (1995), 1628 - 1632.

Simon J. Julier and Jeffrey K. Uhlmann. A General Method for Approximating Nonlinear Transformations of Probability Distributions. http://www.smpp.northwestern.edu/savedLiterature/JulierUhlmannUnscented.pdf, (1996).

Simon J. Julier and Jeffrey K. Uhlmann . Unscented Filtering and Nonlinear Estimation. *PROCEEDINGS OF THE IEEE*, 92 (2004), 401 - 422.

Eric A. Wan and Rudolph van der Menve. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, (2000), 153 - 158.

K. Xionga, H.Y. Zhanga and C.W. Chanb. Performance evaluation of UKF-based nonlinear filtering. *Automatica*, 42 (2006), 261 - 270.

S. A. Banani, and M. A. Masnadi-Shirazi. A New Version of Unscented Kalman Filter. *Engineering and Technology*, 26 (2007), 192 - 197.

Rambabu Kandepu, Bjarne Foss and Lars Imsland. Applying the unscented Kalman filter for nonlinear state estimation. *Journal of Process Control*, 18 (2008), 753 - 768.

Jouni Hartikainen and Simo Sarkka. Optimal filtering with Kalman filters and smoothers : a Manual for Matlab toolbox EKF/UKF. http://read.pudn.com/downloads146/doc/635291/MATLAB_EKF_UKF.pdf, (2007).

Christophe Andrieu, Nando de Freitas, Arnaud Doucet and Michael I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50 (2003), 5-43.

Christopher M.Bishop. Pattern Recognition and Machine Learning, (2006).

D.J.C. MacKay. Introduction to Monte Carlo Methods. *Learning in Graphical Models*, (1998), 175-204.

B.Walsh. Markov Chain Monte Carlo and Gibbs Sampling. *Lecture Notes for EEB 581, MIT*, (2004).

Niclas Bergman. Recursive Bayesian Estimation : Navigation and Tracking Applications. Ph.D thesis, (1999)

Jun S.Liu, Rong Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93 (1998), 1032-1044.

Arnaud Doucet, Simon Godsill and Christophe Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing*, 10 (2000), 197 - 208.

Olivier Cappe, Simon J. Godsill and Eric Moulines. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *PROCEEDINGS OF THE IEEE*, 95 (2007), 899 - 924.

Deok-Jin Lee. Nonlinear Bayesian Filtering with Applications to Estimation and Navigation. Ph.D thesis, (2005)

Arnaud Doucet and Adam M.Johansen. A Tutorial on Particle Filtering and Smoothing : Fifteen years later. http://people.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf, (2008).

A.J. Haug. A Tutorial on Bayesian estimation and Tracking Techniques applicable to Nonlinear and Non-Gaussian Process. *MITRE Technical report*, (2005).

Pierre Del Moral, Arnaud Doucet and Ajay Jasra. Sequential Monte Carlo Sampler. *J.R.Statist.*, 68 (2006), 411 - 436.

N.J. Gordon, D. J. Salmond and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *PROCEEDINGS OF THE IEEE*, 140 (1993), 107 - 113.

Michael Johannes and Nicholas Polson. Particle Filtering. *Handbook of Financial Time Series*,(2009), 1015 - 1029.

Caglar yardim, Zoi-Heleni Michalopoulou and Peter Gerstoft. An overview of Sequential Bayesian Filtering in Ocean Acoustics. *IEEE Journal of Oceanic Engineering*, 36 (2011), 73 - 91.

Fred Daum, and Jim Huang. Particle Degeneracy : Root Cause and Solution. *Signal Processing, Sensor Fusion, and Target Recognition XX*, USA, (2011).

Samarjit Das. Particle Filtering on large dimensional state spaces and applications in computer vision. Ph.D thesis, (2010)

Jinxia Yu, Yongli Tang and Wenjing Liu. Research on Diversity Measure in Particle Filter. *2010 International Conference on Intellugent Computation Technology and Automation*, (2011), 1146-1149.

Peter Bickel, Bo Li and Thomas Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. *IMS Collections, Pushing the Limits of Contemporary Statistics : Contributions in Honor of Jayanta K. Ghosh*, (2008), 318-329.

Christophe Andrieu, Arnaud Doucet and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society*, 72 (2010), 269-342.

Christophe Andrieu, Arnaud Doucet and Roman Holenstein. Particle Markov chain Monte Carlo for Efficient Numerical Simulation. *Monte Carlo and Quasi-Monte Carlo Methods*, (2008), 45-60.

Rudolph van der Merwe, Arnaud Doucet, Nando de Freitas and Eric Wan. The Unscented Particle Filter. *Technical Report, Engineering Department, Cambridge University*, (2000).

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

Francois Bardet and Thierry Chateau. MCMC Particle Filter for Real-Time Visual Tracking of Vehicles. *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, (2008), 539-544.

Thomas Schon, Fredrik Gustafsson, and Per-Johan Nordlund. Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models. *IEEE Transactions On Signal Processing*, 53 (2005), 2279-2289.

Robert Sim, Pantelis Elinas AND James J. Little. A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM. *International Journal of Computer Vision*, 74 (2007), 303-318.

Dieter Fox. KLD-Sampling : Adaptive Particle Filters. *Neural Information Processing Systems*, (2001), 713-720.

Weifeng Liu, Jose C. Principe and Simon Haykin. Kernel Adaptive Filtering : A Comprehensive Introduction., (2010)

S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. Cambridge, MA : The MIT Press, (2005)

F. van der Heijden, R.P.W.Duin, D. de Ridder and D.M.J. Tax. Classification, Parameter Estimation and State Estimation : An Engineering Approach using MATLAB. John Wiley & Sons Ltd, (2004)

Yibing Wang, Markos Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter : a general approach. Transportation Research Part B : Methodological, 39 (2005), 141-167

Bozo Terzic, Martin Jadric. Design and Implementation of the Extended Kalman Filter for the Speed and Rotor Position Estimation of Brushless DC Motor. IEEE Transaction on Industrial Electronics, 48 (2001), 1065-1073

Christophe Andrieu, Johannes Thoms. A tutorial on adaptive MCMC. Stat Comput, 18 (2008), 343-373

Huiying Dong, Yueping Yang. Application of Particle Filter for Target Tracking in Wireless Sensor Network. 2010 International Conference on Communications and Mobile Computing, (2010), 504-508

Carlos M. Carvalho, Michael S. Johannes, Hedibert F. Lopes and Nicholas G. Polson. Particle Learning and Smoothing. Statistical Science, (2010), 88-106

Jonathan Ko and Dieter Fox GP-BayesFilters : Bayesian filtering using Gaussian Processes prediction and observation models. Auton Robot, 27 (2009), 75-90.

Jonathan Ko Gaussian Processes for Dynamic Systems. PhD thesis, University of Washington, (2011).

Jonathan Ko and Dieter Fox Learning GP-BayesFilters via Gaussian Processes latent variable models. Auton Robot, 27 (2009), 75-90.

Neil D. Lawrence Probabilistic non-linear principal component analysis with Gaussian Processes latent variable models. Journal of Machine Learning, 16 (2005), 1783-1816.

Jack M.Wang, David J.Fleet and Aaron Hertzmann Gaussian Process Dynamical Models for Human Motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2008), 283-298.

Jouni Hartikainen and Simo Sarkka. Kalman Filtering and Smoothing Solutions to temporal Gaussian Process Regression Models. 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), (2010), 379-384.

Marc Peter Deisenroth, Marco F.Huber and Uwe D.Hanebeck Analytic Monment-based Gaussian Processes Filtering. Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada (2009), 225-232.

Jan C. Neddermeyer Nonparametric particle filtering and smoothing with quasi-Monte Carlo sampling. Journal of Statistical Computation and Simulation, iFirst, (2011), 1-19.

Carl Edward Rasmussen, and Christopher K.I. Williams. Gaussian Process for Machine Learning. MIT Press, (2006)

Bela A.Frigyik, Amol Kapila and Maya R. Gupta. Introduction to the Dirichlet Distribution and Related Process. Technical Report, Department of Electrical Engineering, University of Washington, (2010)

Francois Caron, Manuel Davy, Arnaud Doucet, Emmanuel Duflos Philippe Vanheeghe. Bayesian Inference for Linear Dynamic Models with Dirichlet Process Mixtures. IEEE Transactions on Signal Processing, 56 (2008), 71-84.

Matthew J.Beal, Zoubin Ghahramani, Carl Edward Rasmussen. The infinite hidden Markov model. Advances in Neural Information Processing Systems, 14 (2002), 577-584.

Yee Whye Teh and Michael I. Jordan. Hierarchical Bayesian Nonparametric Models with Applications. Bayesian Nonparametric : Principles and Practice, Cambridge University Press, (2010)

Emily B. Fox. Bayesian Nonparametric Learning of Complex Dynamical Phenomena. Ph.D thesis, MIT, (2009)

Emily B. Fox, Erik B. Sudderth and Alan S. Willsky Hierarchical Dirichlet Process for Tracking Maneuvering Targets. Proceedings of the 2007 International Conference on Information Fusion, Quebec, Canada, (2007)

Nadine Hilgert, Vivien Rossi, Jean-Pierre Vila and Verene Wagner Identification, Estimation, and Control of Uncertain Dynamic Systems : A Nonparametric Approach. Communications in Statistics - Theory and Methods, 36(2007), 2509-2525.

J. P. de Villiers, S. J. Godsill and S.S.Singh. Particle predictive control. Journal of Statistical Planning and Inference, 141(2011), 1753-1763.

D. Stahl and J. Hauth PF-MPC : Particle filter - Model Predictive Control. Berichte des Fraunhofer ITWM, Nr.201, (2011).

Kiam Choo, David J. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. Proceedings of 8th IEEE International Conference on Computer Vision, 2(2001), 321-328.

Dong Guo and Xiaodong Wang Quasi-Monte Carlo Filtering in Nonlinear Dynamic Systems. IEEE Transactions on Signal Processing, 54(2006), 2087-2098.

Martin Moller Andreasen. Non-linear DSGE Models, The Central Difference Kalman Filter, and The Mean Shifted Particle Filter. CREATES Research Paper 2008-33, University of Aarhus, (2008).

K.J. Friston Variational Filtering. NeuroImage, 41(2008), 747-766.

Tapani Raiko and Matti Tornio. Variational bayesian learning of nonlinear hidden state-space models for model predictive control. Neurocomputing, 72(2009), 3704-3712.

Gareth W. Peters, Geoff R. Hosack and Keith R. Hayes. Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC). http://arxiv.org/PS_cache/arxiv/pdf/1005/1005.2238v1.pdf, (2010).

Nicolas Chopin, Pierre E. Jacob and Omiros Papaspiliopoulos. SMC^2 : A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. http://arxiv.org/PS_cache/arxiv/pdf/1101/1101.1528v2.pdf, (2011).

Nick Whiteley and Christophe Andrieu. Particle Markov Chain Monte Carlo for Multiple Change-point Problems. http://www.maths.bris.ac.uk/research/stats/reports/2009/0911.pdf, (2009).

Yali Wang and Brahim Chaib-draa. KLD-Sampling Particle Markov Chain Monte Carlo for State Estimation. Signal Processing Letters, submitted, (2011).