2013

State Estimation Based on Nested Particle Filters

Swathi Srinivasan
Cleveland State University

STATE ESTIMATION BASED ON NESTED PARTICLES FILTER

SWATHI SRINIVASAN

Bachelor of Technology in Chemical Engineering

Anna University, Chennai, India

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN CHEMICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

August 2013

This thesis has been approved for the

Department of **CHEMICAL AND BIOMEDICAL ENGINEERING**

and the College of Graduate Studies by

Thesis Committee Chairperson, Dr. Sridhar Ungarala

Department/Date

Dr. Jorge Gatica

Department/Date

Dr. Rolf Lustig

Department/Date

Dedicated to my love- Amma and Appa

ACKNOWLEDGMENTS

I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of my thesis.

I would like to express my deep sense of gratitude and thanks to Dr. Sridhar Ungarala, my graduate advisor, the person behind this thesis work, who gave me unrelenting support and incessant encouragement. His willingness to motivate and his patience contributed tremendously to the completion of this work. It is an honor for me to have worked with him. I owe him all my gratitude for everything he has done for me.

I wish to extend my heartfelt thanks to my thesis committee members, Dr. Jorge Gatica and Dr. Rolf Lustig for the guidance they have provided all through my Masters program. They both have played a great role in developing my programming skills.

My earnest gratitude to the Department of Chemical and Biomedical Engineering for their support, especially by Ms. Becky Laird and Ms. Darlene Montgomery. My sincere thanks to my lab mates at the Process Systems Engineering lab for their comments and critics which encouraged me on successful completion.

An honorable mention goes to my family members- Mom, Dad and my loving sister, for their understanding, belief and support. This page cannot end without mentioning the encouragement by friends and brothers who made me realize what I am capable of. I thank them with the whole of my heart for their care and affection.

STATE ESTIMATION BASED ON NESTED PARTICLES FILTER

SWATHI SRINIVASAN

ABSTRACT

In reality many processes are nonlinear and in order to have a knowledge about the true process conditions, it is important to make decisions based on the state of the system. Process measurements such as pressure, temperature, and pH, are available at time instances and this information is necessary in order to obtain the state of the system. Filtering is a state estimation technique by which the estimate is obtained at a time instant, given the process measurements at their respective time instances. Several filters have been developed so far for the estimation of the states of the system. Kalman filters are the optimal filter algorithms used for linear state and measurement models. Approximations are made to this algorithm in order to apply to non-linear systems.

Particle filter (PF) is one such approximation made to the Kalman filtering technique. It involves drawing a set of samples or particles from the state of the system. It works on the principle of importance sampling, where, the samples are derived from a probability density function which is similar to the state model. The particles are resampled according to their weights in order to determine the estimate. Taking into account the difficulties in particle filtering technique, a nested particles

filter (NPF) was developed.

NPF works in such a way that there is a set of particles under each sample of the original particle filter, and from these nest of samples the transition prior is updated using an extended Kalman particle filter (EKPF). The idea of nested particle filter was developed from the unscented particles filter (UPF), which uses the concept of local linearization to develop the importance density. Better importance densities are formulated in this case through which better *posteriori* are obtained. It is important to note that the update of the NPF can be done with any suboptimal nonlinear filter available. This thesis work is based on developing the NPF with a direct sampling particle filter (DSPF) update. Some modifications are made to the unscented particle filter algorithm. The first part of the thesis is to update to NPF with an ensemble Kalman filter update (EnKF). One mathematical example is used to explain the working of the filter and this is compared with the working of NPF with a DSPF update.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS iv

ABSTRACT v

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER

I. INTRODUCTION 1

II. SCOPE OF THESIS 6

2.1 OBJECTIVE 6

2.2 HYPOTHESIS 6

2.3 SPECIFIC GOALS 7

2.4 LAYOUT 8

III. BAYESIAN STATE ESTIMATION 9

3.1 PROBABILITY THEORY 9

3.1.1 Probability 9

3.1.2 Probability of Events 9

3.1.3 Independent Events 10

3.1.4 Conditional Probability 10

3.1.5 Conditional Expectation 11

3.2 BAYES THEOREM 12

3.3 MARKOV PROCESS 13

3.4 STATE ESTIMATION PROBLEM 13

3.4.1 Prediction Stage 15

3.4.2 Update Stage 15
3.5 TYPES OF FILTERS 15
IV. LINEAR FILTERING 18
4.1 KALMAN FILTER 18
4.2 ESTIMATION TECHNIQUE 19
4.2.1 Predict 21
4.2.2 Update 22
V. NONLINEAR FILTERING 23
5.1 Introduction 23
5.2 EXTENDED KALMAN FILTER 24
5.2.1 Predict 25
5.2.2 Update 25
5.2.3 Advantages and Disadvantages 26
5.3 UNSCENTED KALMAN FILTER 26
5.3.1 Unscented Transformation 27
5.3.2 Predict 29
5.3.3 Update 29
5.3.4 Advantages and Disadvantages 30
5.4 PARTICLE FILTERS 30
5.4.1 Sequential Importance Sampling 31
5.4.2 Importance Sampling 32
5.4.3 Degeneracy 34
5.5 DIRECT SAMPLING PARTICLE FILTER 36
5.6 ENSEMBLE KALMAN FILTER 38
5.6.1 Formulation 40
5.6.2 Implementation 40

VI. EXAMPLE FOR NONLINEAR FILTERING 41
VII. NESTED PARTICLES FILTER 46
7.1 BACKGROUND 46
7.2 DRAWBACKS IN CHOOSING THE TRANSITION PRIOR . . 47
7.3 UNSCENTED PARTICLE FILTER 48
7.4 NESTED PARTICLES 49
7.5 ALGORITHM 50
7.6 CHOICE OF THE UPDATE 53
VIII. ANALYSIS OF A NESTED PARTICLE FILTER 54
8.1 NPF WITH EnKF 54
8.2 EFFECT OF THE MEASUREMENT NOISE COVARIANCE . 56
8.3 NPF WITH DSPF 58
8.4 BATCH REACTOR 59
IX. CONCLUSIONS 64
BIBLIOGRAPHY 66
APPENDIX 69
1 Matlab files for nonlinear filter 70
2 Nested particles filter matlab files 78

LIST OF TABLES

Table		Page
I	Sequential Importance Sampling Algorithm	34
II	Performance of the nonlinear filters	42
III	Extended Kalman Particle Filter	50
IV	Unscented Particle Filter	51
V	Performance of a nested particles filter-EnKF	56
VI	Variation of RMSE with respect to measurement noise covariance (constant number of parent particles)	58
VII	Variation of RMSE with respect to measurement noise covariance (constant number of nested particles)	59
VIII	Performance of a nested particles filter -DSPF	59

LIST OF FIGURES

Figure		Page
1	Batch Reactor	4
2	Gaussian pdf	17
3	Non-Gaussian pdf	17
4	Kalman Filter Algorithm	20
5	Unscented Transformation [20]	28
6	Particle Filter algorithm [19]	37
7	Simulation of the state	43
8	System measurement	44
9	Particle Filter- DSPF - EnKF	45
10	Narrowed or peaked likelihood	48
11	Little overlap with likelihood	49
12	Comparison between Particle Filter and NPF	55
13	High measurement noise covariance	57
14	Effect of measurement noise covariance on RMSE	60
15	Nested Particle Filter- DSPF	61
16	NPF- DSPF Batch reactor	63

CHAPTER I

INTRODUCTION

Estimation and filtering are central to a wide variety of disciplines and have their applications in many fields such as control, communication and signal processing, statistics, economics, bioengineering, and operations research. Estimation is a process of arriving at a value for a desired and unknown variable from recent observations and measurements of other variables which are related to desired ones but contaminated with noise. Recently, in the case of engineering, filtering theory has become synonymous with estimation theory. It is odd to note that the word filter is synonymous with estimator, because, in engineering, filters are physical devices that separate the wanted from the unwanted in a mixture. In electronics, filters are named as circuits that are in frequency-selective behavior and select the desired and undesired signals according to the desired input and output. This process of separating the desired and undesired signals is called signal processing. The undesired part of the signal is commonly known as noise. Both the signal and noise are characterized by their covariance functions. Kolmogorov and Wiener used probability distribution to characterize the signal and noise and determine an optimal estimate of the sig-

nal from the given sum of the signal and noise, statistically. In 1960s Kalman used a model in place of the covariance of the signal process which is the fundamental work in linear filtering theory [23]. Kalman developed a filter known as the linear quadratic Gaussian estimation problem. The Kalman filter is also known as the best linear unbiased estimator or the BLUE filter.

Kalman filter (KF) has led to the development of filtering techniques in many ways. With the advent of Kalman filter, the word filtering took a different meaning which is beyond its original literal meaning of separation from mixtures. Thus, the word filter has a variety of functions some of which are:

- ⋆ filter solves an inversion problem where the estimate is obtained from the measured outputs of the system
- ⋆ filter is used as a learning tool by which a certain signal is tracked, estimated, and predicted
- ⋆ filter separates the desired signal from the noise
- ⋆ filter estimates the unknown signal, which is normally a function of state and input of system using measured outputs.

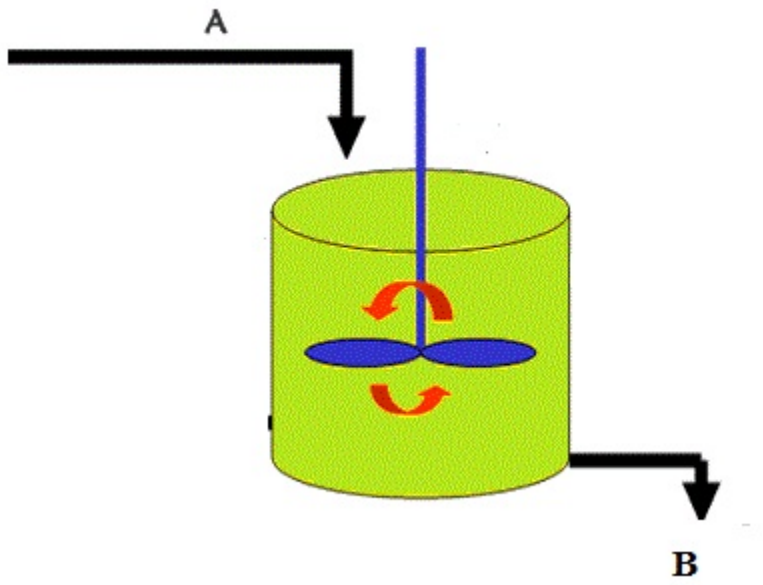
Kalman filters have a wide range of applications in control systems, tracking and navigation of all sorts of vehicles and in the prediction of the future, such as weather forecasting. Other applications of the Kalman filter includes satellite navigation, trajectory estimation, guidance, video and laser tracking systems, radars, and also oil drilling, water and air quality control and geodetic surveys.

In order to understand filtering, consider an isothermal gas phase batch reactor (Fig.[1]). The reactant A forms the product B under isothermal conditions. Once the reactants are fed into the reactor, the concentration or the performance of the reaction is unknown. The desire is to have some knowledge about what is happening

in the reactor. The concentration of the reactants and products may not be measured directly at every moment in the reactor. For example, the dissolved oxygen concentration in bioreactors, temperature in non-isothermal reactors and gaseous flow rates are available for measurement while the values of the concentrations of products, reactants, and biomass are often available only by analysis. So, estimation techniques are used to predict or estimate the variables that are not measured directly. Here, the concentration is the state of the system which is to be estimated. However, pressure sensors are available which give information about the total pressure of the system. This is known as the measurement or observation of the system. In order to compute the concentration, the measurements are used in the state model or the concentration model. A state model is the mathematical relation describing dynamics of the concentration and a measurement model relates the concentrations to the partial pressures of the system. With the knowledge of the measurement and the initial conditions (initial concentration with which the reactor is fed) it is possible to determine the state of the system. This scenario is known as state estimation.

The state of the system is a variable that is used to define the system mathematically. The state variable can be the concentration in the reactors or the flow rates in piping . Similarly, measurements can be pressure, temperature, or pH. The measurements are always accompanied by certain disturbances named as noise. Estimation can be broadly classified into three types based on the time in which the state is evaluated. The measurement data is a set of points obtained at discrete time intervals, which is a function of the state and accompanied by errors or noise. From the measurement set, if the state of the system is computed at the same time instant, then the process is called filtering. If the measurement information is used to compute the state of the system in the past or the future, then it is called smoothing or forecasting, respectively.

Figure 1: Batch Reactor



Kalman filters are applicable for linear systems; however in reality many processes are nonlinear. For example, the concentration dynamics in a batch reactor can be nonlinear depending on the reaction kinetics. Several modifications have been made to the Kalman filtering technique to make it applicable for nonlinear systems.

The probabilistic approach to nonlinear filtering was pioneered by Stratonovich [21]. Kushner [24] and Wonham [25] also developed the continuous nonlinear estimation theory. The probabilistic approach to discrete filtering was developed by Jazwinski [22]. Much of the work on extended Kalman filters (EKF) are one of the earliest approximations available for Kalman filtering technique in order to be applicable for nonlinear systems.The nonlinear state and measurement models are linearized by taking the partial derivatives. The matrix of partial derivatives is known as the Jacobian. When the state and the observation models are highly nonlinear, extended Kalman filters do not hold good. The unscented Kalman filter (UKF) works on the principle of unscented transformation by which a set of sample points called sigma points are

chosen around the mean. These sigma points are then propagated through the state and measurement model and the estimate is obtained. This technique is considered to overcome the disadvantages of extended Kalman filter since it better approximates the nonlinearity and does not involve the calculation of the Jacobian [20].

Particle filter (PF) is based on sequential Monte-Carlo method and is an approximate method for nonlinear filtering. In this method the particles at a given time instant, k, are independent and identically distributed random vectors representing the state vector. These particle/state vectors are used in state equation to find the values of particles at time instant $k+1$ [1]. Particle filter works on the basis of choice of the importance density by a method known as importance sampling. A particle filter with a specific approximation made in the choice of the importance density is termed as bootstrap filtering [5]. If the latest available information is not propagated to the next time instant the particle filter may yield inaccurate state estimates. The disadvantages of the bootstrap filter are taken into account which led to the development of the nested particles filter [18].

The idea of developing a nested particles filter was initiated from the extended Kalman particle filter and the unscented particle filter which used the local linearization concept to obtain the importance density [19]. The unscented Kalman filter is chosen as the importance density for each particle. This choice of importance density allows sampling in a region of high likelihood.

Local linearization technique is used in order to obtain importance densities. These local filters are different types of Kalman filters. Other Gaussian or non-Gaussian pdfs can also be chosen as the local importance density [18]. In this literature, ensemble Kalman filter and direct sampling particle filter are considered as the local importance densities and the resultant superior performance of this estimation technique, when compared to local linearization particle filter is shown.

CHAPTER II

SCOPE OF THESIS

2.1 OBJECTIVE

The main objective of this thesis work is to develop a local linearization particle filter with a different choice of importance densities. Two different suboptimal filters are used as importance densities to obtain the estimate of the state of the system given the observation or the measurements.

2.2 HYPOTHESIS

The approximations made on the choice of the importance density in a particle filter can be overcome by formulating local importance densities on each of the samples. It is hypothesized that the importance density thus generated out of the particles provide accurate information regarding the a posteriori estimate of the state of the system, provided the necessary update method.

2.3 SPECIFIC GOALS

The specific goals or aims considered while achieving objectives of the thesis are listed below. The basic principles involved in the estimation technique is understood using a simple mathematical example and several non-linear filters such as extended Kalman filter, unscented Kalman filter, ensemble Kalman filter, direct sampling particle filter and particle filter are applied to the state estimation problem. Understanding the performance of these filters are very important in order to develop the nested particles filter.

A. Goal 1:- *To investigate nonlinear filters*

Using a simple mathematical example, the non-linear filters can be explained and the performance and efficiency of the filters can be compared with each other. Additionally, simulations are performed to validate the estimations, which can be achieved by obtaining the mean squared error between the simulation and the estimation. Each nonlinear filter has to be compared with each other and their advantages and disadvantages discussed in detail.

The mathematical example is a simple time series system which also has a sinusoidal term to it, which adds to the nonlinearity of the system.

B. Goal 2:- *To investigate the nested particles filter and two update methods*

For the nested particles filtering, it is imperative to understand the logic of unscented particle filters. After the propagation of the nest particles through the state equation, the local importance densities should be obtained for each nest and the optimal importance density is to be achieved. Importance weights are to be evaluated for each nest particle using the Gaussian technique.

The possibilities of using two different update methods are to be analyzed. The DSPF and EnKF methods are to be used in developing the update techniques for the nested particles filter. A comparison is to be made in order to verify which of

the two works better.

C. Goal 3:- *To explain the performance with an example*

An algorithm, a comparison, and a study on the nested particle filter are to be performed with an example. Another example that relates to the concepts of chemical engineering has to be implemented.Proper understanding of the limitations of the nested particles filter are to be ensured along the course of the thesis. A detailed analysis on the limitations of the nested particles filter are to be presented.

2.4 LAYOUT

Chapter 3 explains briefly the concept of Bayesian state estimation, which is important in the filtering problem to obtain the estimate. The algorithm to derive the estimate and the possible assumptions of the best estimate are also explained in this chapter. The classification of filters and the types of filters available are indicated.

Chapters 4 and 5 explain the working algorithm, theory and concept of linear and nonlinear filters respectively. The assumptions and approximations made in obtaining the estimate are explained for each of the filters. The linear Kalman filtering technique and the nonlinear filters such as: EKF, UKF, PF, DSPF and EnKF are explained in these chapters.

Chapter 6 explains a nonlinear system through a mathematical example. The nonlinear filters, PF, DSPF and EnKF are investigated using this mathematical model. A simulation example for the implementation of these filters and a comparison of their performance is also shown.

Chapter 7 and 8 explains the background and algorithm of nested particles filter. The advantages of this technique over the other filters and a comparison of results is shown in these chapters. A comparison between two update methods is explained in Chapter 8.

CHAPTER III

BAYESIAN STATE ESTIMATION

3.1 PROBABILITY THEORY

3.1.1 Probability

Probability is a measure of estimating the likelihood of a statement. Consider an experiment that can produce many results. The collection of all the results is known as sample space.

3.1.2 Probability of Events

Consider a random event whose sample space is S. For each event X of the sample space S, we assume that a number $P(X)$ is defined and satisfies the following conditions:

- $0 \leq P(X) \leq 1$
- $P(S) = 1$

⋆ For a sequence of n events, $X_1, X_2, ...X_n$ that are mutually exclusive, the probability of occurrence of X_1, or X_2, or X_n is given as:

$$p(\cup_{n=1}^{\infty} X_n) = \sum_{n=1}^{\infty} P(X_n) \tag{3.1}$$

The above equation can also be simplified as:

$$p(X \cup Y) = P(X) + P(Y) \tag{3.2}$$

where, X and Y are two mutually exclusive events.

3.1.3 Independent Events

Two events X and Y are said to be independent, if occurrence of one event does not affect the probability of the other.

$$P(XY) = P(X)P(Y) \tag{3.3}$$

3.1.4 Conditional Probability

In probability theory, a conditional probability is the probability that an event would occur based on the condition that another specific event occurs. Let X and Y denote two events. The conditional probability, $P(X|Y)$, is given as, if the probability of event Y occurs, the probability of event X will also occur.

$$P(X|Y) = \frac{P(XY)}{P(Y)}, P(Y) \neq 0 \tag{3.4}$$

where $P(XY)$ is the joint probability distribution. If X and Y are two random variables in the same sample space, the joint probability distribution for X and Y is defined in terms of both X and Y.

3.1.5 Conditional Expectation

If X and Y are discrete random variables, the conditional probability density function of X given that $Y = y$, is defined as,

$$p_{x|y}(x|y) = P\{X = x|Y = y\} \tag{3.5}$$
$$= \frac{P\{X = x, Y = y\}}{P\{Y = y\}} \tag{3.6}$$
$$= \frac{p(x, y)}{p_Y(y)} \tag{3.7}$$

for all values of y such that $p_{Y=y} > 0$.

Similarly, the conditional probability distribution of X given that $Y = y$ is defined as:

$$F_{X|Y}(x|y) = P\{X \leq x|Y = y\} \tag{3.8}$$
$$= \sum_{a \leq x} p_{X|Y}(a|y) \tag{3.9}$$

The conditional expectation of X given that $Y = y$ is given as,

$$E[X|Y = y] = \sum_x xP\{X = x|Y = y\} \tag{3.10}$$
$$= \sum_x xp_{X|Y}(x|y) \tag{3.11}$$

In case of continuous random variables, if X and Y have a joint probability density function $p(x, y)$, then the conditional expectation of X given that $Y = y$ is defined as:

$$E[X|Y = y] = \int_{-\infty}^{\infty} xP_{X|Y}(x|y)dx \tag{3.12}$$

for all values of y such that $P_Y(y) > 0$.

3.2 BAYES THEOREM

Bayesian state estimation is an important technique besed on Bayes theorem [5]. The concept of Bayes theorem can be explained as follows.

Consider two random variables x and y, which are statistically independent. Bayes theorem is based on the fact that joint probability is commutative, that is, $p(x, y) = p(y, x)$. Expanding each side with the conditional probability gives,

$$p(x|y)p(y) = p(y|x)p(x) \tag{3.13}$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{3.14}$$

If the variable x represents the state of the system and the variable y represents the available measurements in the state of the system, then

- $p(x)$ is the probability distribution of the state of the system which is independent of the measurement. This is called the *prior* of x.
- $p(y)$ is the probability distribution of the measurement of the system, which is called the marginal probability or the evidence. It is generally derived as a normalization factor.
- $p(x|y)$ is the probability distribution of the state of the system given the measurements in hand. This is termed as the *posterior* of the system. This is the estimate of the system which is under consideration.
- $p(y|x)$ is the likelihood of the system on the condition that the given model is true.

In Bayesian theory, everything that is unknown is considered as a stochastic

variable. The initial or the prior distribution is assumed to be known from the past or historical data.

The terms in Eq. 3.14 can also be explained as:

$posterior = \frac{(likelihood)(prior)}{evidence}$

In simple words, Bayesian state estimation can be explained as, given the support of, $p(y|x)/p(y)$ and the initial degree of belief, $p(x)$, the *posterior* can be obtained having the measurements in hand.

3.3 MARKOV PROCESS

A Markov process is a stochastic process with the following property. The conditional probability distribution of future states of the process depends only on the present state and not on the sequence of events that preceded it. Consider a stochastic process X_k,(where $k = 0, 1, 2, ...$) with finite set of values. The possible values of the process are denoted by integers{ 0,1,... }. If $X_k = i$, then the process is said to be in state$i$ at time k. Whenever the process is in state i, there is a fixed probability P_{ij} that it will be in state j, that is

$$P_{ij} = P\{X_{k+1} = j | X_k = i, X_{k-1} = i_{k-1}, ..., X_0 = i_0\} \tag{3.15}$$

This is known as the Markov chain. The value of P_{ij} represents the probability that the process when at state i will make a transition into state j.

3.4 STATE ESTIMATION PROBLEM

The Bayesian state estimation problem involves the calculation of the posterior probability density function via the prediction and update stages. The probability density function is assumed to be known. There are a number of optimal and subopti-

mal methods of obtaining the posterior probability density function which is explained in the following sections.

The estimation problems require the state model and the measurement equation model to be known in the probabilistic form. Let the dynamics of the state of the system be modeled as:

$$x_k = f(x_{k-1}, \omega_{k-1}) \tag{3.16}$$

where, x_{k-1} is the state vector at the time instant $k-1$ and ω is the process noise sequence, which is assumed to have a zero mean Gaussian probability density function with known covariance. The initial condition is assumed to be a Gaussian probability density function.

The measurement model is given as:

$$y_k = h(x_k, \nu_k) \tag{3.17}$$

Where, y_k is the measurement at time instant k and ν is the measurement noise sequence, which is assumed to have a zero mean Gaussian probability density function with known covariance. The measurements until time k are available. The initial probability density function of the state vector is available, that is at time instant $k = 0$, called the *prior*. The aim is to find the *posterior* probability density function at time instant k, with the given measurements.

At any time instant k, the state of a system depends only on the value of the previous state $k-1$, according to the Markov property. The above state model is known as the hidden Markov model. The states are hidden and the measurements are available to us.

3.4.1 Prediction Stage

In the prediction state, the state model is used to obtain the *prior* probability density function at time instant k using the Chapman Kolmogorov equation [12]:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{3.18}$$

Here, the term $p(x_k|x_{k-1})$ is a Markov process of order one and the term $y_{1:k-1}$ describes the measurements until the time step $k-1$. The Chapman Kolmogorov equation uses the fact that the state evolution process is a first order Markov process and can be approximated given the state and the measurement model.

3.4.2 Update Stage

At time instant k, the measurement y_k becomes available and Bayes theorem is used to obtain the updated priors. The Bayes rule is given by:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{3.19}$$

Here, the denominator is a normalizing constant given as:

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k \tag{3.20}$$

The measurement at k is used to modify the *prior* so as to estimate the posterior probability density function of the current state. The solution to the above cannot be determined analytically except for linear Gaussian problems. So, various optimal and suboptimal Bayesian methods are used to approximate it.

3.5 TYPES OF FILTERS

Filters can be broadly classified in three different ways, based on the Bayesian solution, the state and the measurement model and the probability distribution func-

tion. One mode of classification of filter is depending on the optimal solution obtained for the filter problem and based on the algorithm. These filters are sub classified into optimal and suboptimal filters. Kalman filters and Grid based filters are examples of the optimal algorithms where a number of assumptions are made to obtain the optimal solution. In the case of Kalman filters, the underlying assumption is that the posterior probability density function is Gaussian, with parameters mean and covariance. In the case of grid based filters, the assumption is that the state space is discrete and contains finite number of states. In many situations, the assumptions made do not hold. These type of filters are called suboptimal filters. Extended Kalman filters and particle filters are examples of suboptimal filters. The second mode of classification is depending on the state and measurement model where the filters are sub-classified as linear and non-linear filters. The third mode of classification is depending on probability distribution function which are sub classified as Gaussian and non-Gaussian filters.

Based on the Bayesian solution:

⋆ Optimal Algorithms:

a) Kalman filters

b) Grid based methods

⋆ Suboptimal Algorithms:

a) Extended Kalman filters

b)Approximate grid based methods

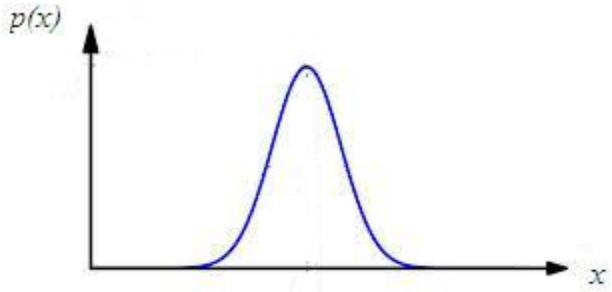
c) Particle filters

Based on the state and measurement model:

⋆ Linear filters

⋆ Nonlinear filters

Figure 2: Gaussian pdf



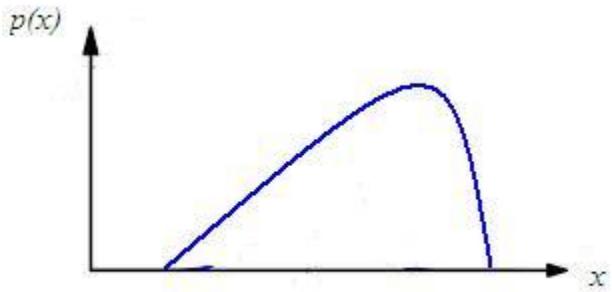
Based on the probability distribution function:

⋆ Gaussian

⋆ Non-Gaussian

Figure 2 shows the Gaussian probability distribution where the mean, median, and mode are the same. Fig. 3 shows the non-Gaussian probability distribution where the mean, median and mode are all different from each other.

Figure 3: Non-Gaussian pdf



CHAPTER IV

LINEAR FILTERING

4.1 KALMAN FILTER

Kalman filtering technique is named after Rudolf Kalman, who developed this theory. Kalman filter is one of the most widely used methods for tracking and estimation of the state of a linear system because it is simple, optimal, and unbiased. Hence Kalman filter is also known as the Best Linear Unbiased Estimator or the BLUE filter.

The Kalman filter is an algorithm that makes the optimal use of imprecise data. It works on a linear system with Gaussian errors and continuously updates the best estimate of the system's current state. This theory is based on a state-space approach where the state and measurement equations are the models of the dynamics and noisy distorted observation respectively.

The state dynamic model is defined as:

$$x_k = Ax_{k-1} + \omega_{k-1} \tag{4.1}$$

The measurement model can be expressed as:

$$y_k = Hx_k + \nu_k \tag{4.2}$$

where,

$$p(\omega_k) \sim N(0, Q) \tag{4.3}$$

$$p(\nu_k) \sim N(0, R) \tag{4.4}$$

x_k is the p dimensional state vector at time k. A is a $p \times p$ dimensional state transition matrix that relates the states of the process at times $k-1$ and k. ω_k is the process noise and it is assumed to be Gaussian (Eq. 4.3), and Q is the covariance matrix or the process noise covariance of dimension $p \times p$. y_k is the m dimensional measurement vector and H is an $m \times p$ dimensional matrix. ν_k is the measurement noise assumed to be Gaussian (Eq. 4.4), and R is the covariance matrix of the measurement noise with dimensions $m \times p$.

4.2 ESTIMATION TECHNIQUE

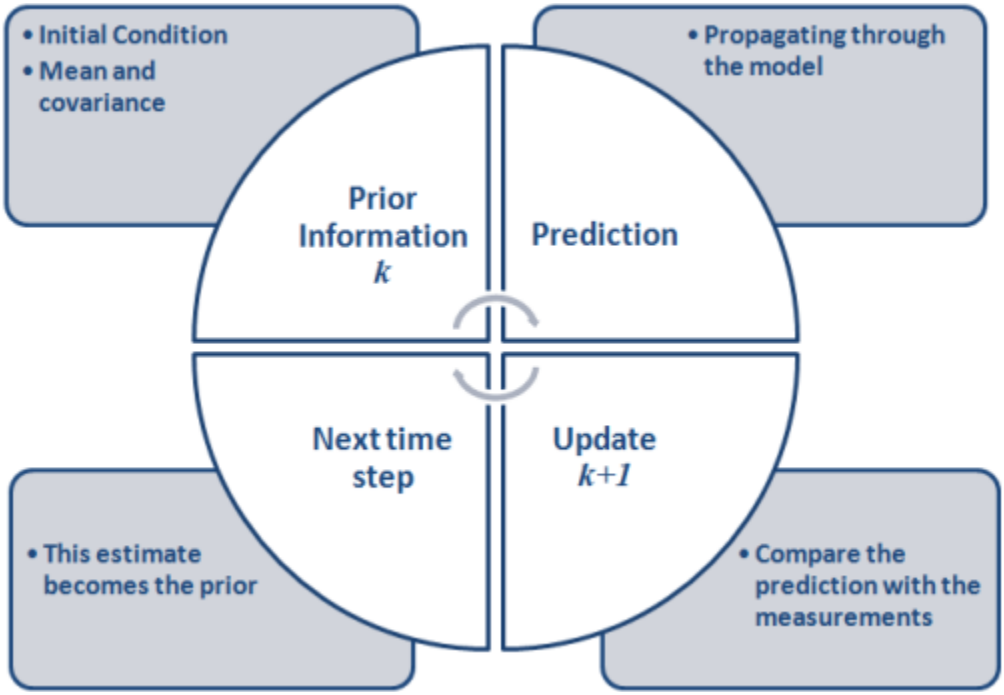
The estimation technique of Kalman filter is recursive, that is, only the previous estimate and the current measurement are required to obtain the current estimate of the system. The algorithm for Kalman filter is best explained in Fig. 4.

The current state estimate can be obtained in the following steps [7]:

- ⋆ propagate the initial condition,
- ⋆ obtain the Kalman Gain,
- ⋆ update the state and covariance estimate.

Consider $\hat{x}_{k-1}$ to be the *prior* state estimate at step k, and then the *posteriori* estimate at step k is $\hat{x}_k$ given the measurement y_k. The *a priori* and *a posteriori*

Figure 4: Kalman Filter Algorithm



state estimate errors are given as:

$$e_k^- = x_k - \hat{x}_k^- \tag{4.5}$$

$$e_k = x_k - \hat{x}_k \tag{4.6}$$

The *a priori* and *a posteriori* estimate error covariances are:

$$P_k^- = E[e_k^- e_k^{-T}] \tag{4.7}$$

$$P_k = E[e_k e_k^T] \tag{4.8}$$

It is desired to obtain the estimate $\hat{x}_k$, given the measurements until the time step k. The measure of the accuracy of the estimate is given by the error covariance matrix P_k. The equations for the Kalman filter are derived in such a way to obtain *a posteriori* state estimate $\hat{x}_k$, as a linear combination of an *a priori* estimate $\hat{x}_k^-$ and a weighted difference between the measurement y_k and a measurement prediction $H\hat{x}_k$.

$$\hat{x}_k = \hat{x}_k^- + K(y_k - H\hat{x}_k^-) \tag{4.9}$$

The difference $(y_k - H\hat{x}_k^-)$ in Eq. 4.9 is called the measurement *innovation* or the *residual* which is the discrepancy between the predicted measurement and the actual measurement. If the residual is zero, then both the predicted and actual measurement are the same.

The matrix K in Eq. 4.9 is known as the *Kalman gain* or the *blending factor* and it minimizes the *a posteriori* error covariance. The Kalman Gain equation is given as,

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{4.10}$$

The Kalman filter estimates the process state at a given time and then obtains the feedback in the form of noisy measurements. Thus it works in the form of a feedback control. The equations for the Kalman filter can be categorized into two groups, the time update equations and the measurement update equations. In the time update or the predict stage, the current state and error covariance estimates are projected forward in time. The resultant is the *a priori* estimates for the next time step. The measurement update equations incorporates the new measurement into the *a priori* estimate and an improved *a posteriori* estimate is obtained.

4.2.1 Predict

The filter is initiated with an initial guess, the initial mean x_0 and the covariance P_0. The predict stage equations are given as

$$\hat{x}_k = A\hat{x}_{k-1} \tag{4.11}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.12}$$

4.2.2 Update

With the predicted measurement available, the measurement at y_k is obtained. The measurement update equations are given in eqs. 4.10 and 4.9. The covariance matrix of the improved posterior estimate is given as:

$$P_k = (1 - K_k H) P_k^- \tag{4.13}$$

The assumptions made in the Kalman filter do not hold true in all situations. Several approximations are made to the Gaussian assumption of the Kalman filter to make it practically applicable to more filter problems. The state estimation problem where the state and the measurement models are nonlinear are explained in the following chapter.

CHAPTER V

NONLINEAR FILTERING

5.1 Introduction

If the state and measurement models are linear and Gaussian, Kalman filter can be directly applied without any arguments. But in reality many processes are nonlinear. The optimal solution of a nonlinear problem can be achieved in a case where all the parameters of a probability distribution are propagated. However in case of a nonlinear system, finite number of parameters are not enough to describe the probability density function [8].The state and the measurement model of the nonlinear system are given as,

$$x_k = f(x_{k-1}) + w_{k-1}(k) \tag{5.1}$$

$$y_k = h(x_k) + v_k \tag{5.2}$$

In the above equations f and h are the state and measurement functions respectively. ω and ν are the state and measurement noise which are assumed to be following Gaussian distribution with the parameters, $(0, Q)$ and $(0, R)$ respectively. The initial

guess of the filter would be x_0 and P_0, which is the mean and the covariance. Here is a mathematical example where the state and the measurement model are nonlinear:

$$x_k = 1 + \sin\left(\frac{\pi(k-1)}{25}\right) + \frac{1}{2}x_{k-1} + w_{k-1} \tag{5.3}$$

$$y_k = \begin{cases} \frac{x_k^2}{2} + v_k & k \leq 30, \\ \frac{x}{2} - 2 + v_k & k > 30. \end{cases} \tag{5.4}$$

The nonlinearity in the above state and the measurement equations are given by the sine term and the square term respectively. The system noise follows a gamma distribution, where as the measurement noise follows a Gaussian distribution. The parameters of these state and measurement noise distributions are described in the following sections.

The following sections describe in detail the underlying assumptions of each nonlinear filter, given the state and measurement model along with the initial conditions.

5.2 EXTENDED KALMAN FILTER

In estimation theory the extended Kalman filter is said to be the nonlinear version of the Kalman filter. The extended Kalman filter approach is to apply the standard Kalman filter to nonlinear systems by performing a linearization around the previous state starting with an initial guess. The linearization procedure is performed while deriving the filter equations and is made on the most recent state reference trajectory using the Taylor series approximation. The state and the measurement models must be differentiable to obtain the first order partial derivative known as the Jacobian.

The predict and the update steps of Kalman filter holds exactly the same,

however the Jacobian is evaluated at every time step in the predict phase. These are used in the Kalman filter equations to linearize the nonlinear state and measurement probability density functions.

5.2.1 Predict

The predict phases are evaluated at the best estimate $\hat{x}_{k-1}$ as in the case of the Kalman filter. Considering a linear Taylor approximation of f at the point $\hat{x}_{k-1}$ and that of h at the point $\hat{x}_k^-$ and ignoring the higher order terms before the first derivative,

$$F_{k-1} = \frac{\partial f}{\partial x}\Big|_{\hat{x}_{k-1}} \tag{5.5}$$

$$H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_k^-} \tag{5.6}$$

where, F_{k-1} and H_k are Jacobian matrices. These matrices are used in the Kalman filter equations.

$$\hat{x}_k^- = f(\hat{x}_{k-1}) \tag{5.7}$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \tag{5.8}$$

5.2.2 Update

The update is exactly the same as that of the Kalman filtering technique. The Kalman gain is evaluated with the predicted covariance and the measurement noise covariance. The following equations correspond to the update step of the extended Kalman filter.

$$\hat{x}_k = \hat{x}_{k|}^- + K_k(\hat{y}_k - H\hat{x}_k^-) \tag{5.9}$$

$$P_k = (I - K_kH)P_k^- \tag{5.10}$$

The Kalman gain is given as,

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{5.11}$$

The term $(H P_k^- H^T + R)$ is known as the innovation or the residual covariance.

5.2.3 Advantages and Disadvantages

The extended Kalman filter is not applicable to systems where the process are inaccurately modeled. The performance of the filter may diverge or lead to suboptimal performance when the model is inaccurate. When the nonlinearity is high in the models, large errors may occur in obtaining the posterior mean and covariance. The poor performance and disadvantages of the extended Kalman filter are explained in detail by Haseltine [6]. Considering all the drawbacks the unscented transformation was developed which overcomes the problems of linearization.

5.3 UNSCENTED KALMAN FILTER

When the state and the observation models are highly nonlinear, that is, the functions f and h, the extended Kalman filter fails to perform. Therefore, a more deterministic and robust technique known as the unscented transform is used in the functioning of the unscented Kalman filter.

The unscented Kalman filter is also known as the Linear Regression Kalman filter, where the nonlinear functions are linearized through linear regression from a set of n points drawn from the *a priori* distribution of the random variable [9]. This technique is proved to be more accurate than the Taylor series linearization since it is derivative free alternative to the extended Kalman filter. The state of the system is represented by a set of points which are termed as the **sigma points**. The unscented

Kalman filter too follows the same steps as that of the Kalman filter, the predict and the update, but selection of sigma points comes first before these steps.

5.3.1 Unscented Transformation

The unscented Kalman filter works on the principle of Unscented Transform, which is given as: *a method for calculating the statistics of a random variable which undergoes a nonlinear transformation* [20]. Consider the same scenario of additive noise terms as in equation 5.1 and 5.2. Fig. 5 shows the concept of unscented transformation. Based on a set of points chosen deterministically the nonlinear function is applied to each and every point and it results in a transformed set of sigma points. Unlike the extended Kalman filter, where linearization is used, the unscented transform shows almost no error in the estimated mean. The predicted covariance matches the true covariance very well.

For instance, consider a random variable x with dimension n. The nonlinear function is defined as $y = f(x)$ with the covariance P_x and mean $\hat{x}$. A set of points known as sigma points are chosen such that their mean and covariance are $\hat{x}$ and P_x respectively. These sigma points are propagated through the nonlinear function $y = f(x)$ to get the $\hat{y}$ and P_y.

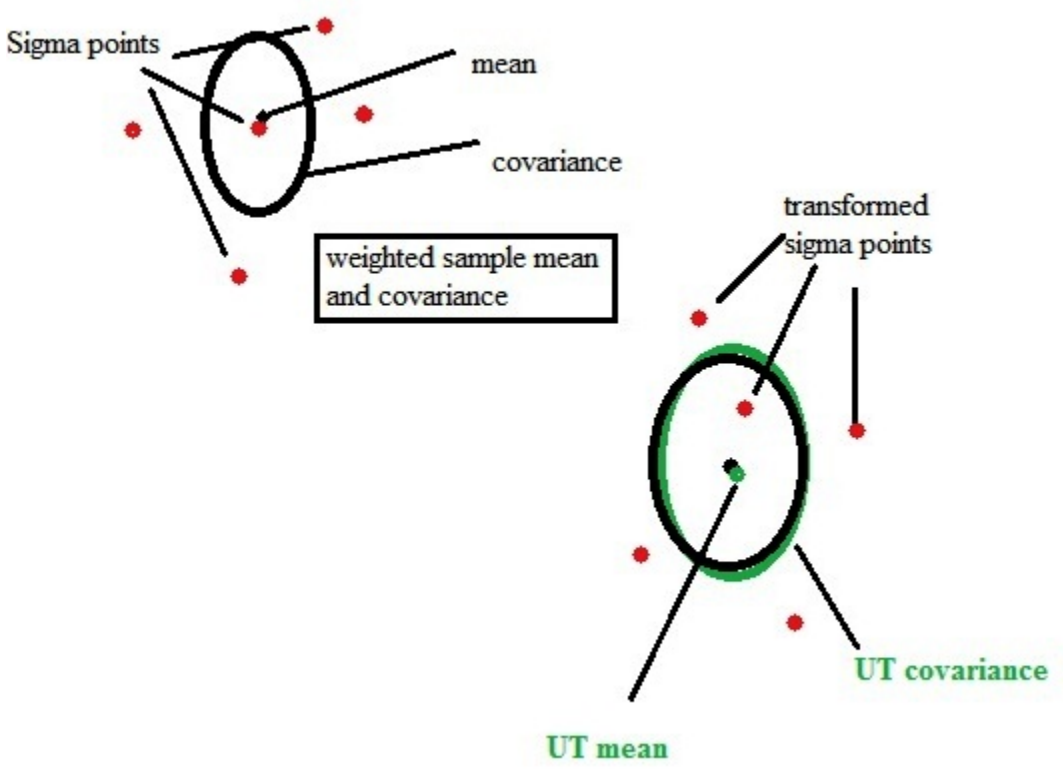
A set of $2n+1$ sigma points (n is the size of the state vector), called as $\mathcal{X}_i$, are selected along with their associated weights W_i. The sigma points are selected as,

$$\mathcal{X}_0 = \hat{x}_0 \tag{5.12}$$

$$\mathcal{X}_i = \hat{x}_0 + (\sqrt{(n+\lambda)P_x})_i \qquad i = 1, ..., n \tag{5.13}$$

$$\mathcal{X}_i = \hat{x}_0 - (\sqrt{(n+\lambda)P_x})_{i-n} \qquad i = n+1, ..., 2n \tag{5.14}$$

Figure 5: Unscented Transformation [20]



The associated weights are evaluated as,

$$W_0 = \lambda/(n+\lambda) \tag{5.15}$$

$$W_0^{(m)} = \lambda/(n+\lambda) + (1-\alpha^2+\beta) \tag{5.16}$$

$$W_i^{(c)} = W_i^{(m)} = 1/\{2(n+\lambda)\} \qquad i = 1, ..., 2n \tag{5.17}$$

where, $\lambda = \alpha^2(n+k) - n$ is a scaling parameter. α is the spread of the sigma points which is a small positive number in general. κ is another scaling parameter set to 0. β is equal to 2 in case of Gaussian distributions. $W_i^{(m)}$ and $W_i^{(c)}$ are the weights of the mean and covariance associated with the i^{th} point.

These sigma points are propagated through the nonlinear function,

$$\mathcal{Y}_i = f(\mathcal{X}_i) \qquad i = 0, ..., 2n \tag{5.18}$$

The mean and covariance of the nonlinear transformation are obtained from these sigma points.

$$\hat{y} \approx \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_i \tag{5.19}$$

$$P_y \approx \sum_{i=0}^{2n} W_i^{(c)} \{\mathcal{Y}_i - \hat{y}\}\{\mathcal{Y}_i - \hat{y}\}^T \tag{5.20}$$

5.3.2 Predict

The prediction step in the unscented Kalman filter involves a set of $2n + 1$ sigma points calculated from the previous known mean x_{k-1}. These sigma points are then propagated through the nonlinear state equations to find the predicted state and covariance matrix ($\hat{x}_k^-$ and $\hat{P}_{k-1}^-$) given as,

$$\hat{x}_k^- = \sum_{i=0}^{2n} W_i^m \chi_k^{i-} \tag{5.21}$$

$$P_k^- = \sum_{i=0}^{2n} W_i^c [\chi_k^{i-} - \hat{\chi}_k^-][\chi_k^{i-} - \hat{\chi}_k^-]^T \tag{5.22}$$

The sigma points are calculated again at this point.

5.3.3 Update

The sigma vectors are propagated through the nonlinear function,

$$\mathcal{Y}_k^i = h(\chi_{k|k-1}^i), i = 0, ...2n \tag{5.23}$$

The predicted measurement and the covariance matrix are obtained,

$$\hat{y}_k = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_k^i \tag{5.24}$$

$$P_{y,k} = \sum_{i=0}^{2n} W_i^c [\mathcal{Y}_k^i - \hat{y}_k][\mathcal{Y}_k^i - \hat{y}_k]^T \tag{5.25}$$

The state and the measurement cross covariance matrix is,

$$P_{x,y,k} = \sum_{i=0}^{2n} W_i^c [\chi_{k|k-1}^i - \hat{\chi}_{k|k-1}][\mathcal{Y}_k^i - \hat{y}_k]^T \tag{5.26}$$

Finally, the Kalman gain, the updated state estimate and the covariance are given as:

$$K_k = P_{y,k} P_{y,k}^{-1} \tag{5.27}$$

$$\hat{x}_k = \hat{x}_k + K_k(y_k - \hat{y}_k) \tag{5.28}$$

$$P_k = P_k^- - K_k P_{y,k} K_k^T \tag{5.29}$$

5.3.4 Advantages and Disadvantages

Compared to the extended Kalman filter, the unscented Kalman filter predicts the state of the system more accurately and it is easier to implement. The unscented Kalman filter can also consider the noise models to be non-Gaussian or non-additive. The number of sigma points can be extended to increase the efficiency of the filter. However, the UKF involves a complex computations and can be applicable only to models with Gaussian noises.

5.4 PARTICLE FILTERS

Particle filter works on the principle of sequential Monte-Carlo method for solving recursive Bayesian problems. A set of random samples called particles are drawn

from a set of sequential probability distributions. These particles are propagated over time using two techniques: importance sampling and resampling. As the number of particles increases to infinity a more accurate solution to the estimation problem is obtained and the posterior density function becomes more accurate. If the number of particles is large it increases the computational complexity, hence the number of particles is restricted. There are a number of particle filtering algorithms, for example, sequential importance sampling, sampling importance resampling, auxiliary sampling importance resampling and regularized particle filter [1]. The particle filter algorithm is applicable when the process and/or the measurement noise is non-Gaussian. It does not require the linearization technique of the state and the measurement models.

5.4.1 Sequential Importance Sampling

The sequential importance sampling algorithm is also known as the bootstrap filtering algorithm. It involves implementing a Bayesian filter by Monte-Carlo simulations [5].

The posterior density function is represented by a set of N random samples with associated weights. Consider a set of random samples, $\{x^i_{0:k}, w^i_k\}^N_{i=1}$ that denote the posterior density $p(x_{0:k}|y_{1:k})$, where $\{x^i_{0:k}, i = 0, ..., N\}$ is a set of support points having the weights $\{w^i_k, i = 1, ..., N\}$. The weights are normalized such that $\sum^N_{i=1} w^i_k = 1$. The posterior density is approximated as:

$$p(x_{0:k}|Y_{1:k}) \approx \sum_{i=1}^{N} w^i_k \delta(x_{0:k} - x^i_{0:k}) \tag{5.30}$$

δ is the Dirac delta function. The weights are chosen using the importance sampling technique. Understanding the particle filter algorithm involves understanding the concept of importance sampling [10].

5.4.2 Importance Sampling

Importance sampling is a technique for estimating the properties of a particular distribution by generating samples from a different distribution than the pdf of interest. It is assumed that it is difficult to draw samples from the probability density of interest and hence another density is used from which samples are drawn. The importance sampling principle is described as follows: suppose $p(x)$ is a probability density from which it is difficult to draw samples, consider another density $\pi(x)$ which is easily evaluated, on the condition $p(x) \propto \pi(x)$. The N samples are drawn from another density $q(x)$, which is called the *importance density*. The importance density is a kernel function, which is used to express the importance weights in a recursive form. The associated weights, are given as [10],

$$w^i \propto \frac{p(x^i)}{q(x^i)} \tag{5.31}$$

where, $q(x^i)$ is the probability density function value at x^i. Then:

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i) \tag{5.32}$$

Sequentially, at each iteration, the samples $x^i_{0:k}$ are available and the next time step is evaluated with the samples in hand. The weights are updated by,

$$w^i_k \propto \frac{p(x^i_{0:k}|y_{1:k})}{q(x_{0:k}|y_{1:k})} \tag{5.33}$$

The importance density is chosen such that,

$$q(x_{0:k}|y_{1:k}) = q(x_k|x_{0:k-1}, y_{1:k})q(x_{0:k-1}|y_{1:k-1}) \tag{5.34}$$

The posterior density can be calculated as,

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_k|x_{0:k}, y_{1:k-1})p(x_{0:k}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{5.35}$$

$$= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|y_{1:k-1})} \times p(x_{0:k-1}|y_{1:k-1}) \tag{5.36}$$

$$\propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|y_{1:k-1}) \tag{5.37}$$

The weight update equation is calculated as:

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{0:k-1}^i|y_{1:k-1})}{q(x_k^i|x_{0:k-1}^i, y_{1:k})q(x_{0:k-1})^i|y_{1:k-1})} \tag{5.38}$$

$$= w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, y_{1:k})} \tag{5.39}$$

In case of a bootstrap particle filter, the transition density is considered to be the importance density, one kind of an assumption in choosing the importance density [5]. The assumption is:

$$q(x_k|x_{0:k-1}, y_{1:k}) = q(x_k|x_{k-1}, y_k) \tag{5.40}$$

Therefore the weights become:

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tag{5.41}$$

Thus the posterior density becomes:

$$p(x_k|y_{1:k}) \approx \sum_{i+1}^{N} w_k^i \delta(x_k - x_k^i) \tag{5.42}$$

The choice of an importance density is very important for obtaining the good results from a particle filter. The support points of the posterior pdf must coincide with the support points of the chosen importance density. The importance density must be chosen in such a way that it minimizes the variance of the measurement noise. Table I explains the algorithm for the sequential importance sampling particle filter [1].

Table I: Sequential Importance Sampling Algorithm

1. Initialization - Assume set of N random samples (particles) $\{x^i_{k-1} : i-1, ..., N\}$ from the conditional probability density function: $p(x_{k-1}\|y_{1:k-1})$
2. Prediction - Propagate N values $\{v^i_{k-1} : i = 1, ..., N\}$ from the density function of process noise v_{k-1} Generate new sample points $\{x^i_{k\|k-1} : i = 1, ..., N\}$ using: $x^i_{k\|k-1} = f(x^i_{k-1}, v^i_{k-1})$
3. Update - with the measurement y_k assign each a weight of $x^i_{k\|k-1}$ The calculated weight: $w^i_k = \frac{p(y_k\|x^i_{k\|k-1})}{\sum_{i=1}^N p(y_k\|x^i_{k\|k-1})}$ The posterior probability distribution is given as $p(x_k\|y_k) = \sum_{i=1}^N w^i_k \delta(x_k - x^i_{k\|k-1})$

5.4.3 Degeneracy

One of the difficulties in implementing the sequential importance sampling particle filters is the degeneracy problem. In case of a particle filter, after a number of iterations,all but one particle will have negligible weight. Thus, with time, the estimation of the probability distribution becomes very poor and the solution to update the particles become trivial. The degeneracy is related to the variance and is measured using the effective sample size N_{eff}.

$$N_{eff} = \frac{N}{1 + Var(w^i_k)} \tag{5.43}$$

An approximate value of the above equation is obtained by taking the inverse of the sum of the squared weights:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w^i_k)^2} \tag{5.44}$$

When N_{eff} is less than N then the degeneracy is severe. The degeneracy

problem can be solved by choosing a good importance density and by resampling.

Choice of good Importance Density:

The importance density is chosen in such a way that it minimizes the variance of the importance weights, to reduce the degeneracy problem. The variance becomes zero when the importance density is equal to the transition density:

$$q(x_k|x^i_{0:k-1}, z_{0:k}) = p(x_k|x^i_{k=1}, y_k) \tag{5.45}$$

The weight equation reduces to:

$$w^i_k \propto w^i_{k-1} p(y_k|x^i_k) \tag{5.46}$$

Resampling:

Whenever a significant degeneracy arises it can be reduced by the resampling technique. The basis of resampling is to eliminate the particles with small weights and choose particles with larger weights. The resampling step involves generating a new set of samples by resampling a number of times, with replacement. There are a number of resampling algorithms, out of which the most commonly used ones are listed below [2].

- ⋆ Multinomial resampling
- ⋆ Residual resampling
- ⋆ Stratified resampling

Multinomial Resampling:

A set of N uniform random numbers are generated and they are used to select x_k according to multinomial distribution. That is,

$$x_k = x(F_{-1}(u_k)) = x_i \tag{5.47}$$

where,

$$u_k = u_{k+1}\tilde{u}_k^{\frac{1}{k}} \tag{5.48}$$

with $\tilde{u}_k \sim U[0,1)$ representing the uniform random numbers. Here, F_{-1} is the generalised inverse of the cumulative probability distribution of the normalised particle weights.

Residual Resampling:

In this method for $i = 1, ..., N$, the number of samples is given by the following equation,

$$N = \lfloor nw^i \rfloor + N^i \tag{5.49}$$

where $\lfloor \rfloor$ denotes the integer part and is distributed according to multinomial distribution and n is the number of particles that are resampled.

Stratified Resampling:

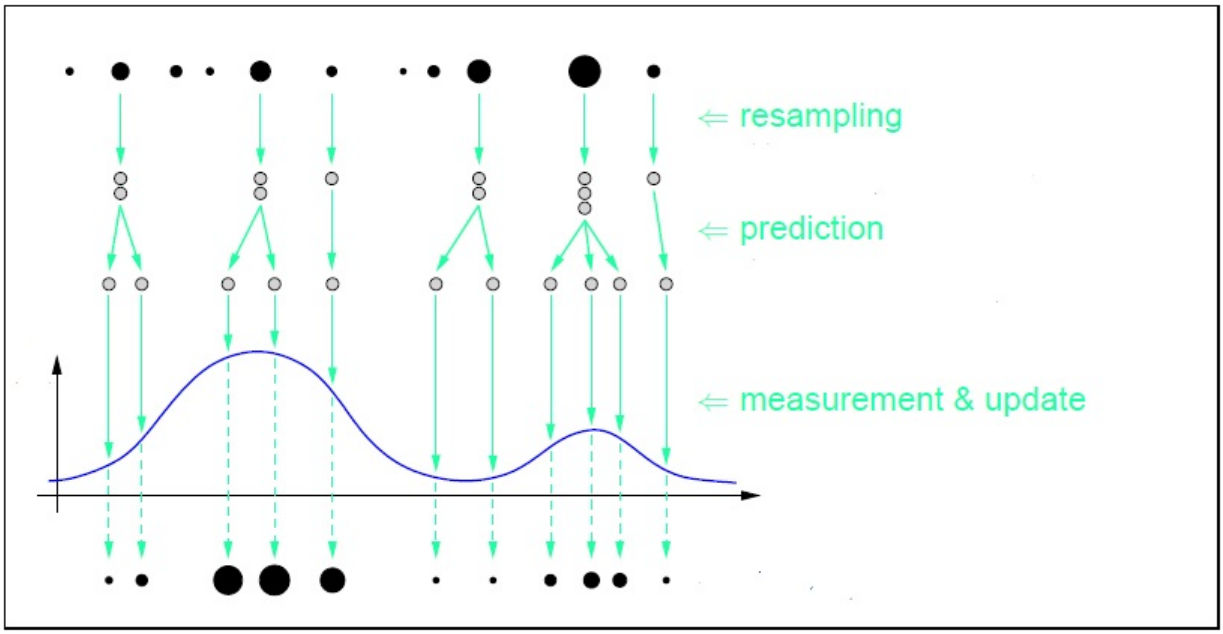
Stratified resampling is based on survey sampling and consists of pre-partitioning the $(0, 1]$ interval into n disjoint sets. The inversion method as in multinomial resampling is used.

The advantages and disadvantages of a particle filter are explained in the following chapter. Fig. 6 shows the representation of a particle filter algorithm. Initially, the samples are with equal weights (not shown in figure). After the importance sampling, the particles are assigned with their respective weights. Resampling is done, where the particles with low weights are discarded and replaced with other particles.

5.5 DIRECT SAMPLING PARTICLE FILTER

A direct sampling particle filter was developed for a constrained state space problem, and works on obtaining the estimate from an approximate conditional prob-

Figure 6: Particle Filter algorithm [19]



ability density. It can also be applied to unconstrained systems and is proved to be working better than a particle filter under certain conditions [16]. In case of a particle filter, if the Gaussian assumptions on the noise terms are not taken into consideration, it is difficult to draw samples. Whereas, in case of a direct sampling particle filter, the Gaussian assumptions on prior is taken into consideration and samples are drawn directly from the approximate conditional density. The approximate conditional density is a normalizing product of the *a priori* probability density and the likelihood function given by $p(y_k|x_k)$.

$$p^*(x_{k+1}|y_{1:k+1}) = \frac{1}{c} e^{-1/2(||x_{k+1}-\hat{x}_{k+1}||^2_{P^-_{k+1}{}^1} + ||y_{k+1}-h(x_{k+1})||^2_{R^{-1}})} \tag{5.50}$$

where c is a normalizing constant and p^* is the constrained conditional probability density.

The algorithm for an unconstrained direct sampling particle filter is shown

below:

- ⋆ Samples are generated as in the case of a particle filter and they are propagated through the system model to obtain the predicted samples, $\{\hat{x}^i_{k+1}\}$.
- ⋆ The mean and the covariance are computed for the predicted samples, given as $\hat{x}_{k+1}$ and P_{k+1} respectively.
- ⋆ A bank of N candidate samples are drawn from the support using the boundaries of a 3-sigma rule.
- ⋆ The normalized values of approximate conditional density, $p^*(x_{k+1}|y_{1:k+1})$ is evaluated over the candidates.
- ⋆ Conditional samples are drawn from the bank of candidate samples according to the approximate conditional density and the discrete cdf.
- ⋆ The mean and covariance of the posterior are computed from the samples.

The direct sampling particle filter works better than a particle filter for a constrained as well as unconstrained system, having both linear and nonlinear constraints. However in some cases, the Gaussian approximation leads to a poor performance compared to a particle filter [16].

5.6 ENSEMBLE KALMAN FILTER

The ensemble Kalman filters (EnKF) are generally a Monte Carlo approximation of Kalman filters which are applicable for systems where the covariance matrix is replaced by a sample covariance. The ensemble Kalman filter can be compared to a particle filter in one way, where the ensembles or samples of an EnKF are similar to that of the particles of a particle filter. The underlying assumption in an EnKF is

similar to that of a Kalman filter where the probability distributions are considered to be Gaussian. Consider a set of ensembles,

$$X = [x^1, ..., x^N] = [x^i] \tag{5.51}$$

where, X is an $n \times N$ matrix whose columns are a sample from the prior distribution. The matrix X is called the *prior ensemble*. The measurement or the observation matrix (also called the data) is replicated as a $m \times M$ matrix so that each column consists of the measurement vector plus the random vector from the normal distribution $N(0, R)$, where R is the covariance matrix. The mean of the ensemble and the sample covariance are the prior estimates in case of an EnKF.

The measurement data,

$$Y = [y_1, ..., y_N] = [Y_i] \tag{5.52}$$

$$\hat{X} = X + K(Y - HX) \tag{5.53}$$

The above forms a random sample set from the posterior distribution. The ensemble Kalman filter is obtained by replacing the state covariance matrix P in the Kalman gain matrix, $K = PH^T(HPH^T + R)^{-1}$ by the sample covariance computed from the ensemble and it is called the *ensemble covariance*. The Kalman gain is evaluated at the update step from the sample covariance.

5.6.1 Formulation

The ensemble mean and covariance for the above mentioned state and measurement matrix are given as,

$$E(X) = \frac{1}{N}\sum_{i=1}^{N} x^i \tag{5.54}$$

$$C = \frac{AA^T}{N-1} \tag{5.55}$$

where,

$$A = X - E(X) = X - \frac{1}{N}(Xe_{N\times 1})e_{1\times N} \tag{5.56}$$

C is the sample covariance and e is the matrix of all ones of the indicated size.

The posterior ensemble is given as,

$$\hat{X} \approx X^i = X + CH^T(HCH^T + R)^{-1}(Y - HX) \tag{5.57}$$

The posterior ensemble consists of linear combinations of members of the prior ensemble. R is the covariance matrix and it is always positive.

5.6.2 Implementation

A function $h(x) = Hx$ is called the observation or the measurement function. The posterior ensemble is rewritten as,

$$X^i = X + \frac{1}{N-1}A(HA)^T P^{-1}(Y - HX) \tag{5.58}$$

where,

$$HA = HX - \frac{1}{N}((HX)e_{N\times 1})e_{1\times N} \tag{5.59}$$

and

$$P = \frac{1}{N-1}HA(HA)^T + R \tag{5.60}$$

The ensemble update is computed by evaluating the observation function h on each ensemble member once. For a large number of measurement or observation data points, the ensemble Kalman filter can be applicable with the modifications made [4].

CHAPTER VI

EXAMPLE FOR NONLINEAR FILTERING

In order to understand the working of the nonlinear filters, consider the following mathematical example which is a benchmark problem in the field of state estimation. The state and the measurement models are given as,

$$x_k = 1 + \sin(\frac{\pi(k-1)}{25}) + \frac{1}{2}x_{k-1} + w_{k-1} \tag{6.1}$$

$$y_k = \begin{cases} \frac{x_k^2}{2} + v_k & k \leq 30, \\ \frac{x}{2} - 2 + v_k & k > 30. \end{cases} \tag{6.2}$$

The state model in the above equation has a sinusoidal term which is nonlinear. Also the measurement model has a switch term in it, which provides two set of measurements. The state and measurement noise terms are additive, which provides randomness to the system.

The initial conditions of the filter are given as x_0 is 1 and the covariance P_0 is 1. The system noise follows a Gamma distribution with parameters $(3, 0.5)$ and the measurement noise follows a Gaussian distribution with parameters $(0, 10^{-5})$. With

Table II: Performance of the nonlinear filters

Filters	MSE×100	STD×100	CPUtime in ms
PF	50.5	14.92	0.55
DSPF	41.17	14.28	0.43
ENKF	394.88	107.87	0.018

this information in hand, the true simulation of the state and the measurement model can be seen in Figs. 7 and 8.

Using this example EnKF, particle filter and DSPF nonlinear filters are illustrated. The performance of the filters are shown separately and the obtained Mean Square Error values are also shown in Table II. MSE is a measure of the error between the simulated and the estimated values.

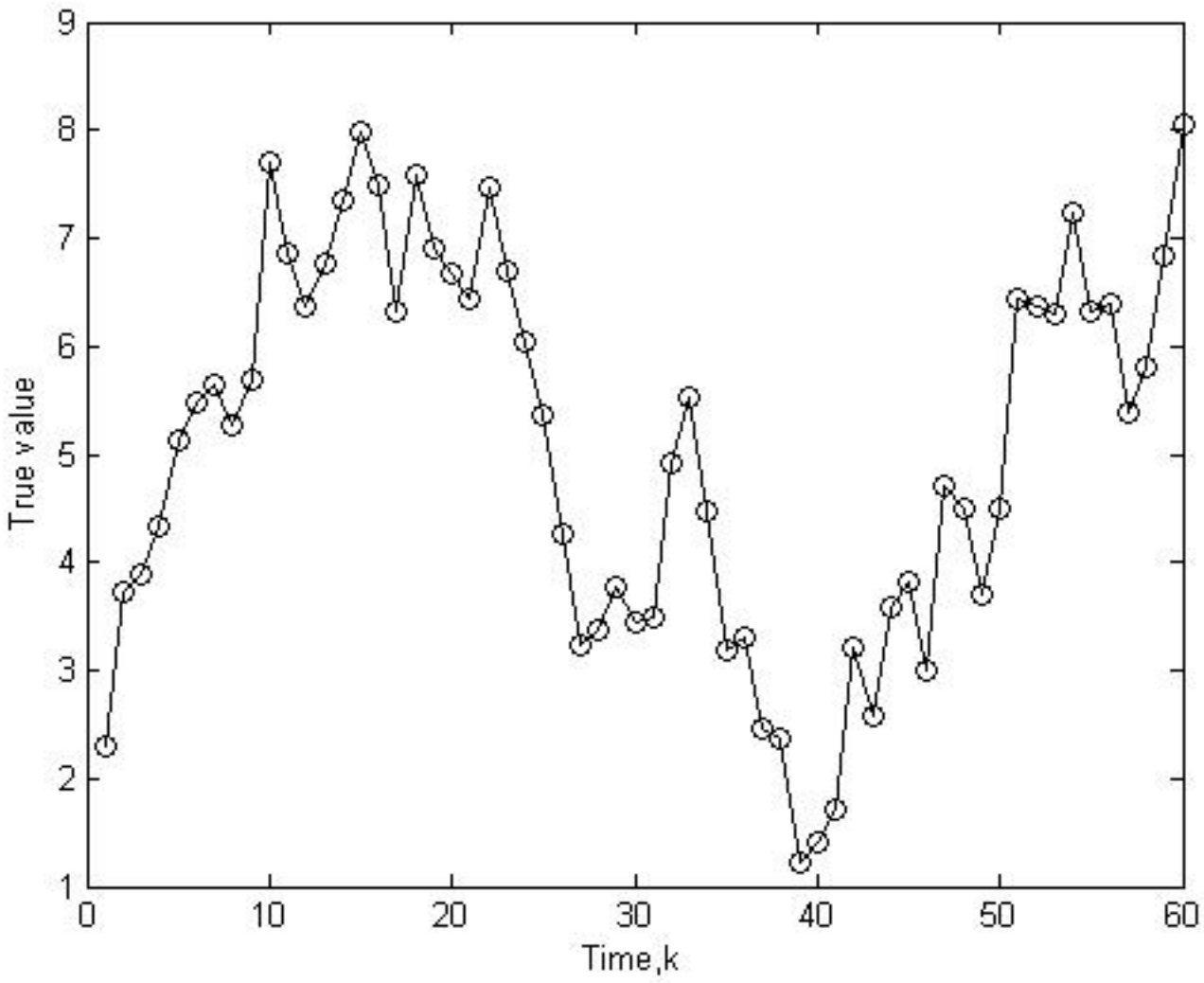
$$MSE = \frac{1}{Kn}\sum_{k=1}^{K}(x_k - \hat{x}_k)^T(x_k - \hat{x}_k) \tag{6.3}$$

where K is the number of realizations considered. The MSE values are averaged over 100 realizations. In case of the particle filter, DSPF and the EnKF, the sample size was chosen as 200. To achieve a proper comparison, the estimated values and the true values are plotted together and the extent of coincidence between them is observed.

The simulation plot in the Fig. 7 shows the state of the system at each of the 60 time steps. The estimates obtained from each of the method is compared with this number and the mean square error between these values are estimated. This is the meassure for the filter performance.

Table II shows the comparison between the performances of a particle filter, a direct sampling particle filter and an ensemble Kalman filter. The standard deviation of the MSE values are also displayed in the table. The CPU time denotes the

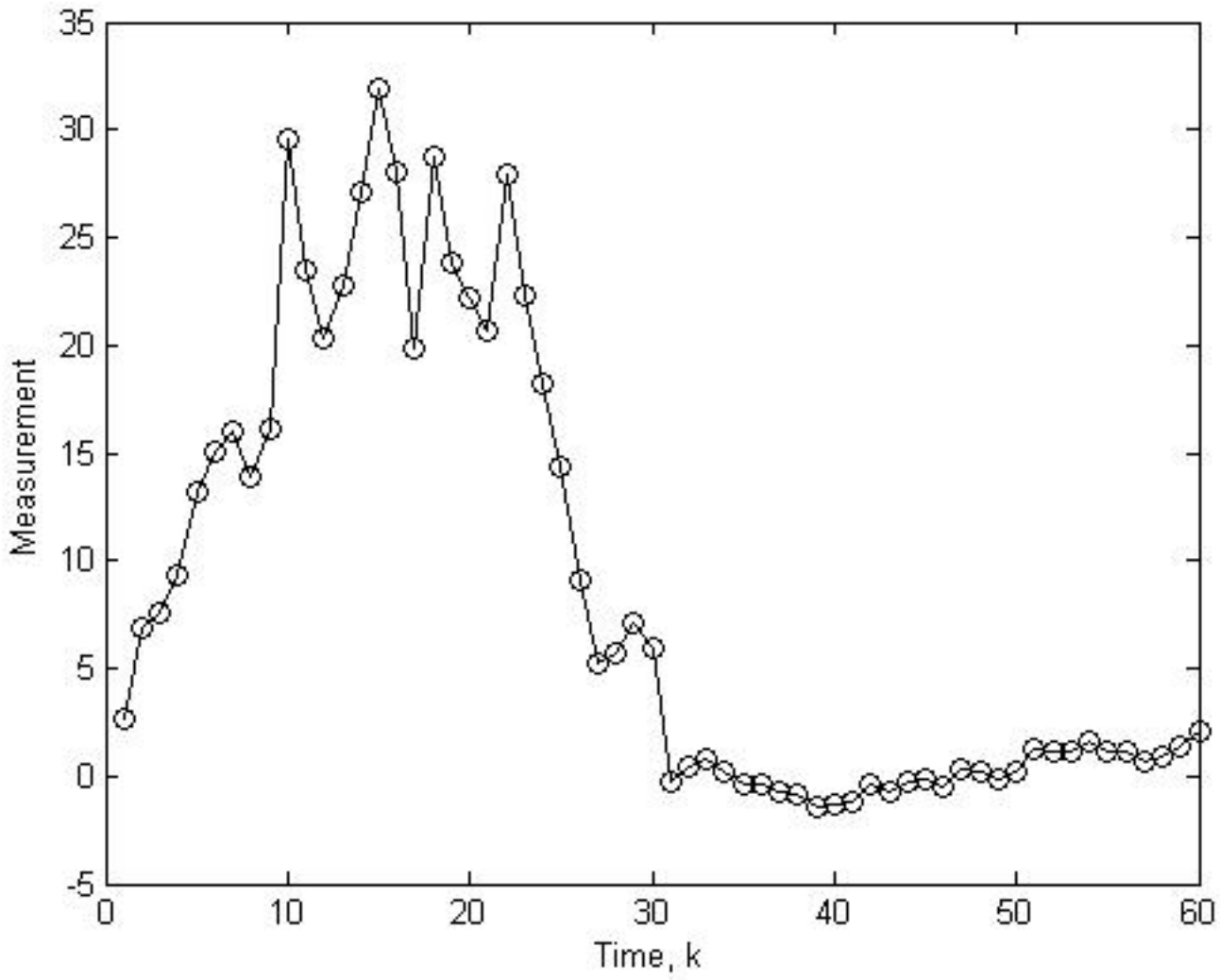
Figure 7: Simulation of the state



computational time taken by Matlab to compute each of the filtering algorithms on a 2.30 GHz processor. Since the linear Kalman filter approximations hold in the ENKF it shows poor performance compared to the particle filter and DSPF. It is clear that the DSPF works better than a particle filter in this case as discussed in the previous chapters.

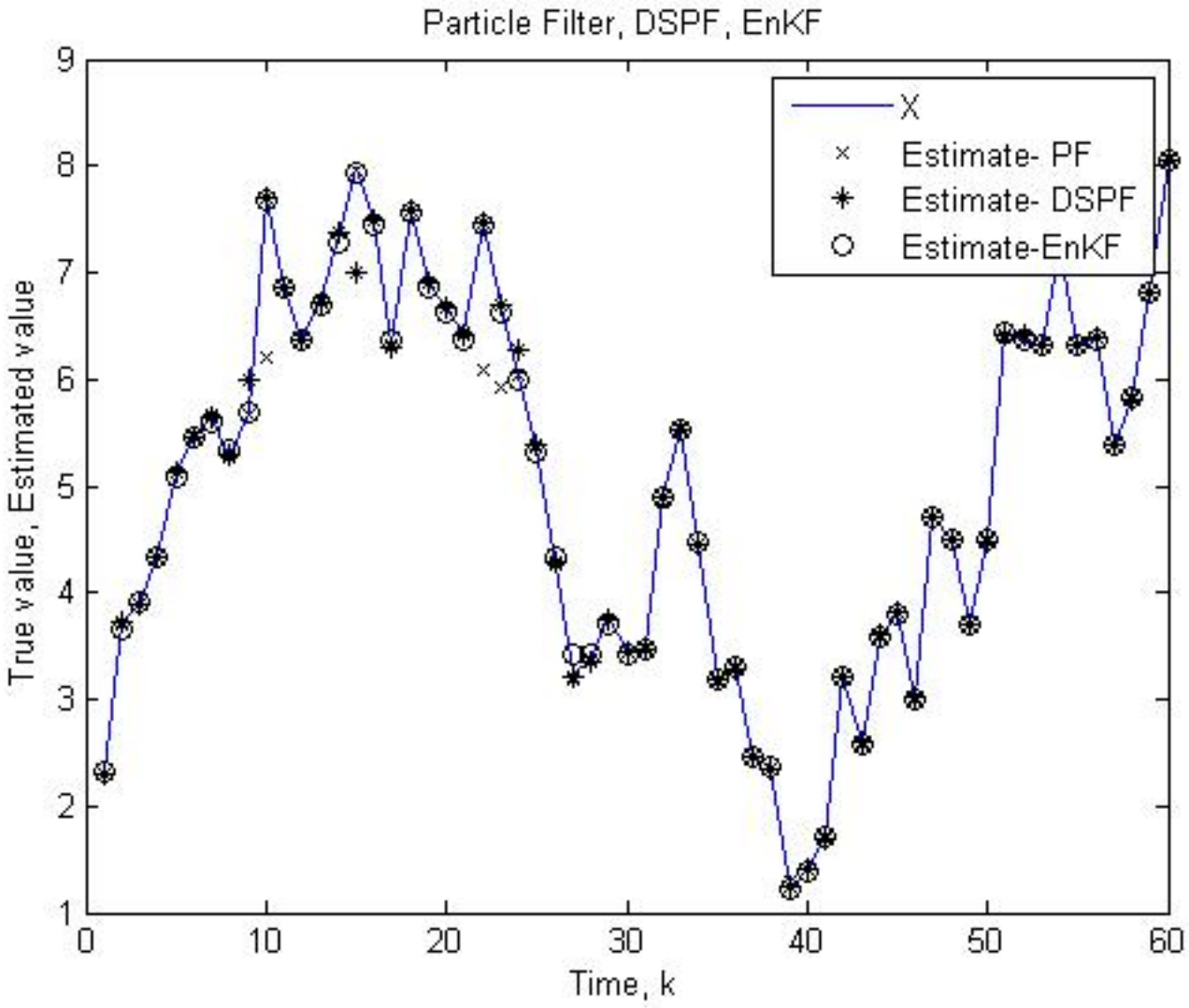
Fig. 9 represents the true simulation values and the estimated values of the state of the system.

Figure 8: System measurement



From the table and the figures, it is seen there are a number of factors which influence each of the filters and it is difficult to justify the difference or the poor performance of each of the filter with just one parameter. The sample size, the covariance and the random numbers generated are all important in influencing the performance of a filter. However, under certain conditions, a comparison between these filters can be made. For instance, in the above example, the measurement and the process noise covariances are kept the same for each of the filter. The sample

Figure 9: Particle Filter- DSPF - EnKF



size is also kept constant in all three of the filters. Under these conditions, the direct sampling particle filter and the ensemble Kalman filter's performance is greater than the particle filter.

CHAPTER VII

NESTED PARTICLES FILTER

7.1 BACKGROUND

The idea of the nested particles filter first originated with the unscented particles filter [19]. This algorithm involves the unscented Kalman filter (UKF) to generate the importance density for the particle filter. The UKF uses the most recent measurement to obtain the importance density for each of the samples in the particle filter. The availability of importance densities leads to better performance of the particle filter. This led to the development of unscented particle filter due to shortcomings of the extended Kalman filter and the unscented Kalman filter.

The application of the ensemble Kalman filter or the direct sampling particle filter is not the only idea behind the development of nested particles filter. It also revolves around updating the group of particles branched under one particular parent particle, called the nest, along with the parent particles. The resampling method in the particle filter discards the unwanted samples and replicates the samples according to its weights. Here, the same method is employed to both the parent particles and

group of particles branched under the parent particles which will be explained later.

The choice of the importance density plays a vital role for particle filters. A special case, where the importance density is chosen to be the transition prior has several drawbacks to it. The generalized particle filter, or the bootstrap particle filter is the type of the particle filter, that chooses the transition prior to be the importance density on the condition that they lie on the same support points as those of the posterior density.

7.2 DRAWBACKS IN CHOOSING THE TRANSITION PRIOR

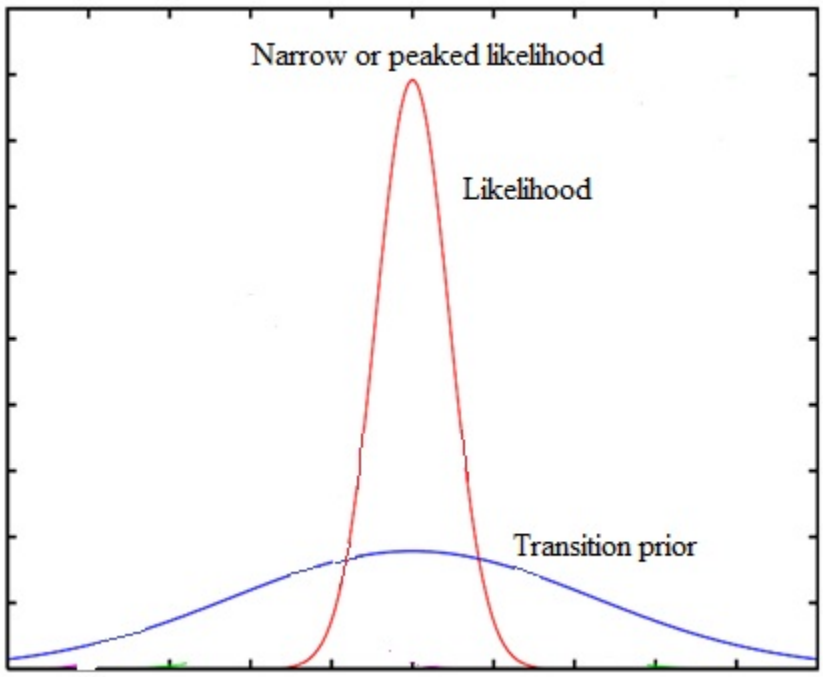
The choice of the transition prior as the importance density has two major problems concerning the likelihood.

- ⋆ *Narrow or peaked likelihood* In Fig. 10, the likelihood is narrow compared to the transition prior, which explains the area of overlapping between the two regions. Thus the area of overlapping is less which is called narrow or peaked likelihood.

- ⋆ *Little overlap with likelihood* In Fig. 11 the likelihood and transition prior have an area of intersection, this explains the overlapping to be less compared to narrow of peaked. Thus it is called little overlap with likelihood.

There are different ways to overcome the drawbacks. One way to choose better transition prior is to enhance the choice of importance density. With no information about what the likelihood is, it is difficult to arrive at a conclusion about the optimal choice of an importance density. The nested particles filter was developed in order to overcome these shortcomings of a particle filter.

Figure 10: Narrowed or peaked likelihood



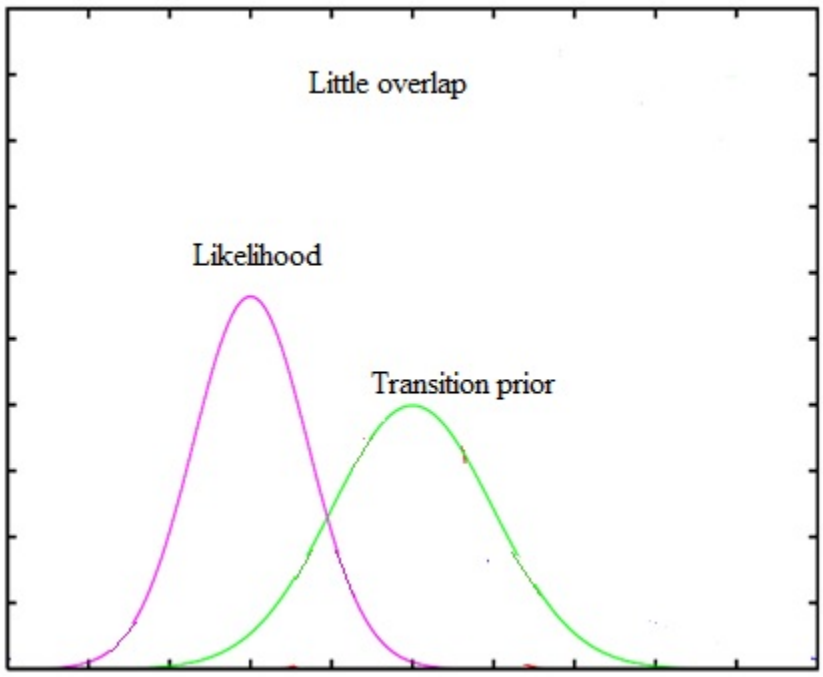
7.3 UNSCENTED PARTICLE FILTER

As mentioned in earlier sections, using the transition prior as proposal distribution is inefficient for certain cases. It is important to shift the particles to a region of higher likelihood. In order to achieve this, the unscented Kalman filter is used as the proposal density or the importance density.

The new filter that results from using an extended Kalman filter for proposal density within the particle filter framework is called extended Kalman particle filter shown in Table III. Similarly if the unscented Kalman filter is used as the proposal density in the particle filter, then the resulting filter is called unscented particle filter, shown in Table IV.

This method used here results in a better sampling algorithm [19], where the variance of each proposal distribution changes with time. The extended Kalman filter obtains an estimated value closer to the likelihood, when 0the underlying Gaussian

Figure 11: Little overlap with likelihood



assumption still remains the same. The unscented particle filter was developed to overcome some of the drawbacks of the extended Kalman particle filter.

The unscented Kalman filter propagates the mean and covariance of the Gaussian approximation to the state distribution more accurately compared to that of the extended Kalman particle filter. More accurate estimates of the true covariance are obtained from the unscented Kalman filter. This filter generates distributions which provide a more solid support or there is more overlap between the true posterior distribution and the distributions generated by the unscented Kalman filter.

7.4 NESTED PARTICLES

A set of N random samples are considered which are distributed according to the posterior density. These particles are known as the parent particles of the system. Consider another set of N particles which individually contain M number

Table III: Extended Kalman Particle Filter

1. **Initialization** - The particles x_0^i are drawn from the prior $p(x_0)$
2. **Importance Sampling** - - Compute the Jacobians of the process and the measurement models - The particles are updated with the EKF algorithm - The importance weights are evaluated - The importance weights are normalized
3. **Selection** - The particles are multiplied with their respective importance weights to obtain the N random particles
4. The final estimate is similar to the sequential importance sampling particle filter

of nest particles under each of them. The nested particles are related to the parent particles in such a way that the samples mean is the parent particle.

$$x_{k+1|k+1}^i = \hat{x}_{k+1|k+1}^i = \frac{1}{M}\sum_{j=1}^{M} x_{k+1|k+1}^{ij} \tag{7.1}$$

The N nested particles generate the local importance densities from the measurement information available at the given time step. The nest particles are propagated in time forward with the state of the system. The nest is further updated with the measurement and any suboptimal filter can be used at this stage for updating. The nested particles are now used to evaluate the local importance density from which the weights are evaluated and further resampling is done for both parent and nest particles.

7.5 ALGORITHM

The algorithm for the nested particles filter is explained in this section. The first step in the nested particles filter estimation is to initialize the parent particles which are sampled from the probability density function of the state of the system,

Table IV: Unscented Particle Filter

1. **Initialization** - The particles x_0^i are drawn from the prior $p(x_0)$
2. **Importance Sampling** - - Compute the Sigma points - The particles are propagated into the posterior using the time update steps of the UKF - The new observation or the measurement update is obtained - The sample and the set are both updated - The importance weights are computed and normalized
3. **Selection** - The particles are multiplied with their respective importance weights to obtain the N random particles
4. The final estimate is similar to the sequential importance sampling particle filter

such that $\{x_{k|k}^i : i = 1, ..., N\}$. For each of the parent particle, the child nests are sampled in such a way that, $\{x_{k|K}^{ij} : 1, ..., M\}$. The initial conditions are given at time instant $k = 0$.

At this stage, a nest consisting of the parent particle and the corresponding child particles are obtained. This setup can be visualized as a matrix of the order $M \times N$. The next step is to propoagate the nest particles using the system equation and the random samples generated by the noise parameters.

$$x_{k|k-1}^{ij} = f(x_{k|k}^{ij}) + w_{k-1}^{ij} \tag{7.2}$$

The measurement nest is computed using the measurement equation and the noise parameters.

$$y_{k|k-1}^{ij} = h(x_{k|k-1}^{ij}) + v_k^{ij} \tag{7.3}$$

The child particles are updated using the measurement equation. This update is similar to the Kalman filter algorithm, which results in the updated particles as follows:

$$x^{ij}_{k|k} = x^{ij}_{k|k-1} + T^{i}_{k|k-1}(S^{i}_{k|k-1})^{-1}(y_k - \mathbf{y}^{ij}_{k|k-1}) \tag{7.4}$$

where, $S^{i}_{k|k-1}$ is the innovation covariance or the covariance matrix of the measurement model. $T^{i}_{k|k-1}$ is the cross covariance between the process and the measurement distributions.

The next step is the choice of the importance density. The importance density chosen is the Gaussian importance density, which is given by,

$$\pi^{i}(x_k|x_{k-1}, Y_k) \approx \mathcal{N}(x_k : \hat{x}^{i}_{k|k}, P^{i}_{k|k}) \tag{7.5}$$

The parent particles are sampled from the importance density and the importance weights are computed using

$$\tilde{w}^{i}_{k} \propto \tilde{w}^{i}_{k-1} \frac{p(y_k|x^{i}_{k|k})p(x^{i}_{k|k}|x_{k-1})}{\pi^{i}(x_k|x_{k-1}, Y_k)} \tag{7.6}$$

The importance weights are normalized,

$$w^{i}_{k} = \frac{\tilde{w}^{i}_{k}}{\sum_{i=1}^{N} \tilde{w}^{i}_{k}} \tag{7.7}$$

The parent particles are resampled according to the importance weights and particles are replicated or discarded according to the weights being low or high. The corresponding nests are also replicated or discarded along with the parent particles.

The child particles are shifted to a new mean value with respect to the new resampled parent particles. The mean of the parent particles are computed, which is the final estimate.

$$x^{ij}_{k|k} = x^{ij}_{k|k} - \hat{x}^{ij}_{k|k} + x^{i}_{k|k} \tag{7.8}$$

$$\hat{x}_{k|k} = \frac{1}{N} \sum_{i=1}^{N} x^{i}_{k|k} \tag{7.9}$$

7.6 CHOICE OF THE UPDATE

In the above algorithm, the update method used for updating the parent particles can be modified and a different update technique can be used. The ensemble Kalman filter and the direct sampling particle filter are the two choices considered for the update of the parent particles. The basic algorithm of both these filters were explained in the previous chapters. The extent to which the choice of the update method influences the nested particle filter performance is analyzed and the limitations of using this nested particle filter with the two update method is discussed in detail in the following chapter.

CHAPTER VIII

ANALYSIS OF A NESTED PARTICLE FILTER

8.1 NPF WITH EnKF

The nested particles filter has been formulated for the same time series example described in Chapter VI. The filter is initialized with the initial conditions $x_0 = 1$ and $P_0 = 1$. The estimation is run for one hundred realizations and the average of the root mean square error (RMSE) values are determined.

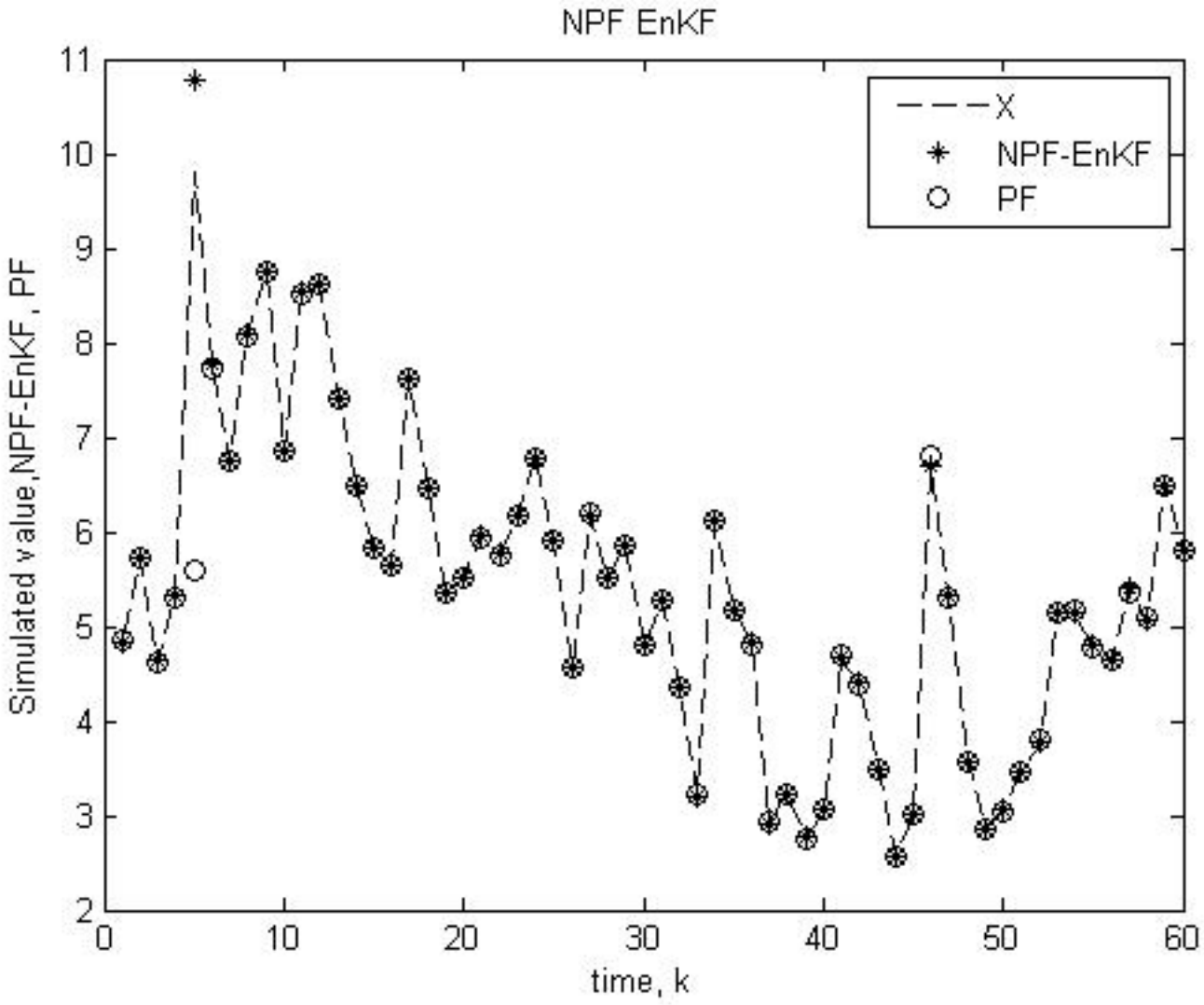
The RMSE is defined as,

$$RMSE = \sqrt{\frac{1}{Kn}\sum_{k=1}^{K}(x_k - \hat{x}_k)^T(x_k - \hat{x}_k)} \tag{8.1}$$

Fig. 12 shows a comparison between the particle filter and the nested particle filter that uses an ensemble Kalman filter for the update of the parent particles.

In Fig. 12, the particle filter and the nested particle filter both are very close to the simulated values. The sample size is one among the number of factors that

Figure 12: Comparison between Particle Filter and NPF



influence the performance of the filter. The number of parent particles and nest particles are important to achieve more accurate posterior density functions. In the above situation, the particle filter is run at a particle size of 200, whereas the nested particle filter was run at 200 parent particles with 50 child particles for each parent.

The performance of a nested particles filter with an EnKF update is shown in Table V. The ensemble Kalman filter is run at different sample sizes and the effect of the sample size on the estimate is evaluated.The number of the parent particles varies

from 25 to 500. Similarly the number of the nest child particles varies from 25 to 200. The RMSE value and the standard deviation is obtained for each of the case. As the sample size increases, a general trend of increase in accuracy of the estimation is seen. The RMSE value tends to decrease with an increase in the number of particles.

Table V: Performance of a nested particles filter-EnKF

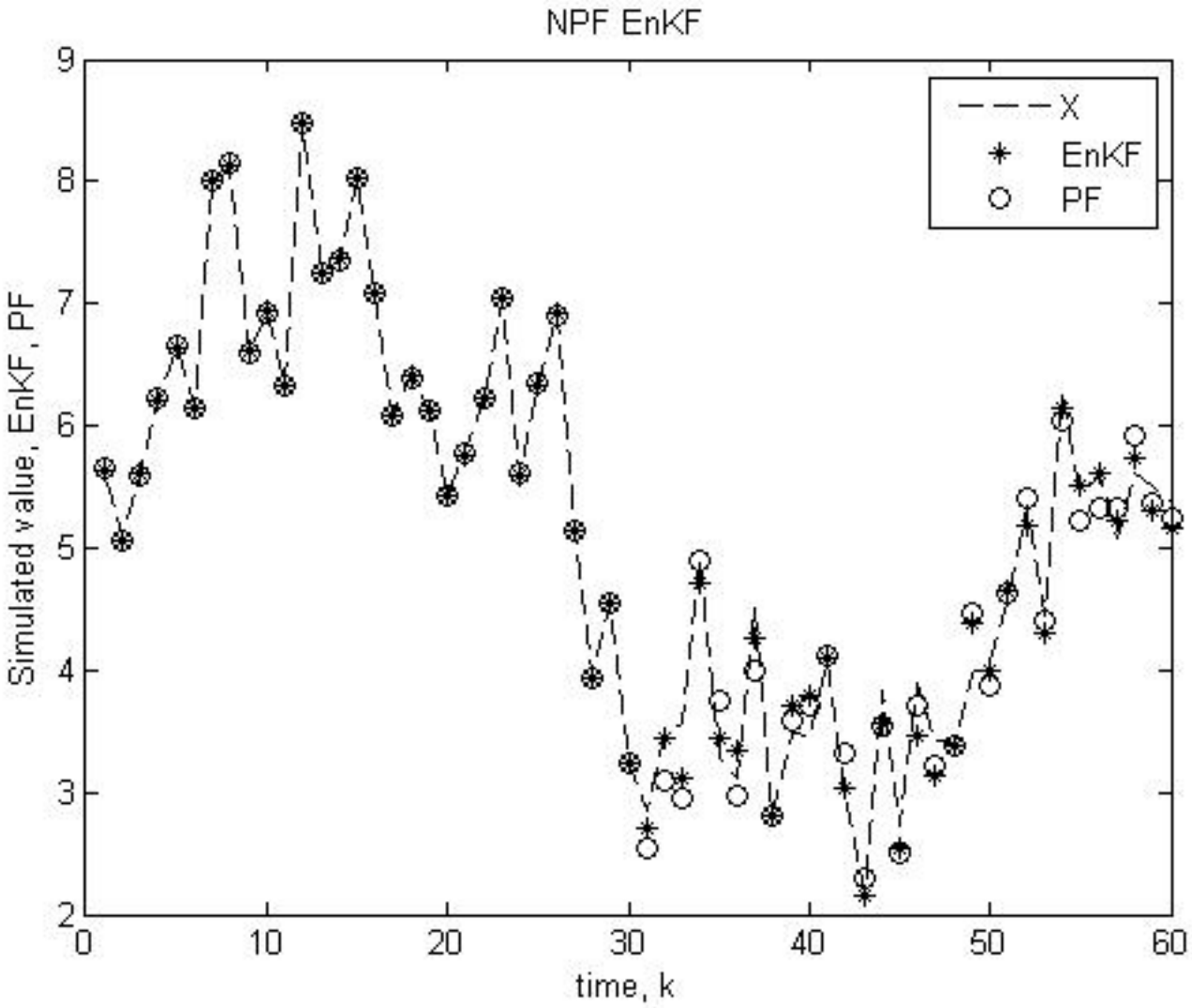
Filter	**RMSE**×100 **(STD**×100**)**				
NPF- EnKF	N=25	N=50	N=100	N=200	N=500
M=25	4.09 (1.04)	3.09(0.68)	2.45(1.04)	1.92(0.62)	1.18(0.48)
M=50	3.7(0.73)	3.25(0.64)	2.65(0.76)	1.45(0.56)	1.01(0.32)
M=100	3.51(0.59)	2.83(0.56)	2.58(2.25)	2.2(0.62)	0.49(0.28)
M= 200	3.69(0.37)	3.13(0.63)	2.32(0.81)	1.27(0.26)	0.35(0.19)

Fig. 13 is an example of the poor performance of an nested particle filter, where the measurement noise covariance is set a high value, 0.01. The plot shows that some of the estimated values do not match or overlap with the true simulated values, which will increase the RMSE. If the measurement noise covariance is high, it means that there is very less confidence placed on the system. Under such situations the nested particles filter loses its accuracy. The estimated values are not close to the simulated values.

8.2 EFFECT OF THE MEASUREMENT NOISE COVARIANCE

An analysis is made by varying the measurement noise covariance values for different particle sizes. The original value for the measurement noise covariance is 1×10^{-5}. This measurement noise covariance value is increased in steps from 1×10^{-5} to 1. The RMSE value in accordance to each of the specified particle size are observed. Tables VI and VII show the variation in the RMSE value with respect to the size of

Figure 13: High measurement noise covariance



the parent particle as well as the size of the nest particle. First the number of parent particles are fixed and the number of nested child particles is varied, and the process is reversed.

Things to be noted from the two tables is that, as the measurement covariance value increases, to some extent, the RMSE value decreases. This trend breaks after some point, which makes it significant that at higher sample size and high measurement noise covariance, with very high values of the RMSE. This is illustrated in Fig.

Table VI: Variation of RMSE with respect to measurement noise covariance (constant number of parent particles)

	RMSE ×100					
Covariance	**0.00001**	**0.0001**	**0.001**	**0.01**	**0.1**	**1**
PF	51.54	36.39	28.48	25.44	48.26	73.19
M = 25	1.92	2.53	4.99	13.83	54.23	185.57
M = 50	1.45	2.35	4.79	13.92	53.3	178.88
M = 100	2.2	2.02	5.44	13.94	54.19	175.92
M = 200	1.27	2.7	5.13	14.61	53.2	180.55

14, where R denotes the measurement noise covariance. The sample size and the confidence placed on the measurement plays an important role in the performance of a nested particles filter.

It can also be seen that, up to a certain particle size and variance, the RMSE value decreases but suddenly it increases after further increase in the particle size. This can be interpreted as an indication that an optimum particle size is achieved in such cases.

8.3 NPF WITH DSPF

The nested particles filter can be updated with different suboptimal filters. The direct sampling particle filter is another choice that can be made in order to update the nested particles filter. The same example as discussed in the previous sections is considered here. Fig. 15 is the outcome of the performance of the nested particle filter with a DSPF update equation. Comparing Figs. 15 and 12, the algoritm that uses the DSPF update equation seems to yield more accurate estimates, i.e., closer to the simulated values. This is further supported by Table VIII. Comparing the

Table VII: Variation of RMSE with respect to measurement noise covariance (constant number of nested particles)

	RMSE ×100					
Covariance	**0.00001**	**0.0001**	**0.001**	**0.01**	**0.1**	**1**
PF	51.54	36.39	28.48	25.55	48.26	73.19
N = 25	3.69	3.27	5.72	15.1	46.94	116.45
N = 50	3.13	2.8	5.27	14.68	48.51	130.08
N = 100	2.32	2.85	5.01	14.64	52.24	155.78
N = 200	1.27	1.9	4.82	14.61	53.2	180.55

results of an EnKF with the DSPF the DSPF tends to perform more accurate than the EnKF, when the sample size is low.

In general, it is observed that, the more number of particles the better the estimate is. However, large sample sizes leads to computational complexity [13]. When a nested particles filter is coupled with the DSPF better results are achieved compared to the ensemble Kalman filter.

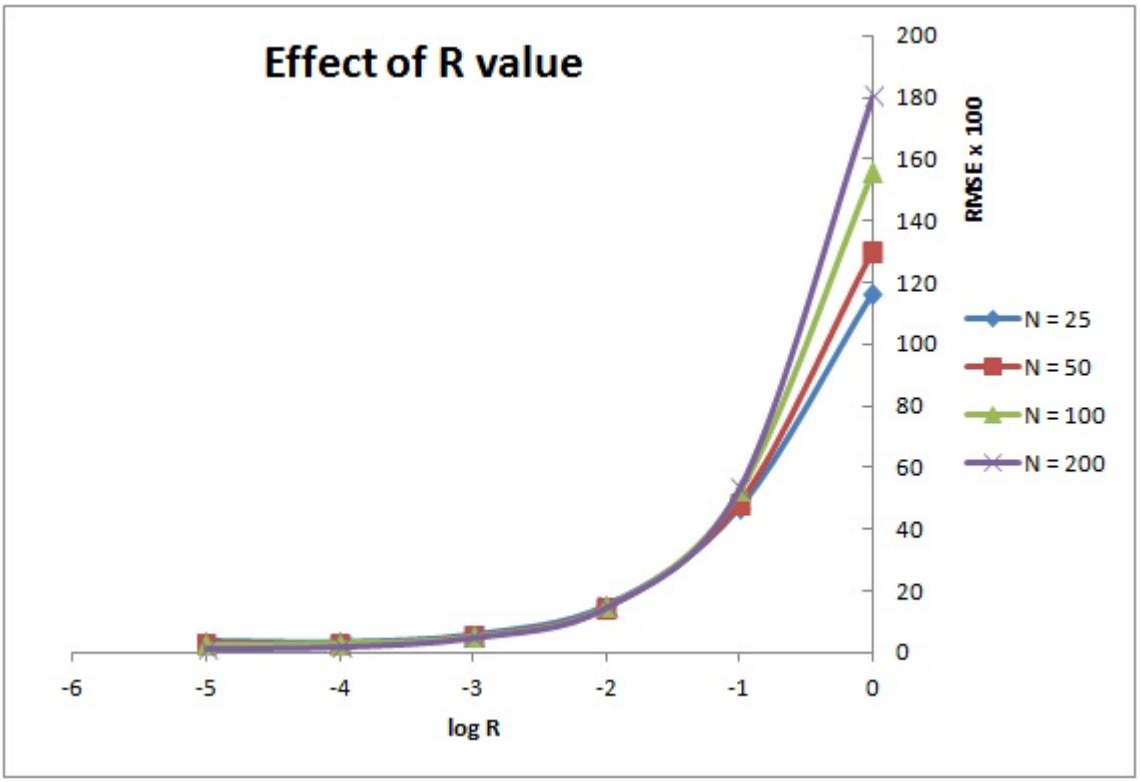
Table VIII: Performance of a nested particles filter -DSPF

Filter	**RMSE**×100 (**STD**×100)				
NPF- DSPF	N=25	N=50	N=100	N=200	N=500
M=25	3.33 (0.98)	3.29(0.78)	2.14(1.03)	1.52(0.62)	1.08(0.54)
M=50	3.19(0.84)	3.15(0.59)	2.72(0.84)	2.15(0.56)	0.91(0.39)
M=100	2.51(0.47)	2.34(0.46)	1.58(1.15)	1.3(0.62)	0.39(0.18)
M= 200	2.69(0.22)	2.13(0.32)	1.32(0.51)	1.27(0.26)	0.25(0.09)

8.4 BATCH REACTOR

Consider a three state batch reactor whose reversible reactions are given by,

Figure 14: Effect of measurement noise covariance on RMSE



$$A \rightleftharpoons B + C \tag{8.2}$$

$$2B \rightleftharpoons C \tag{8.3}$$

The rate constants of the forward reactions are given by k_1, k_3 respectively and the rate constants of the backward reactions are k_2, k_4 respectively.

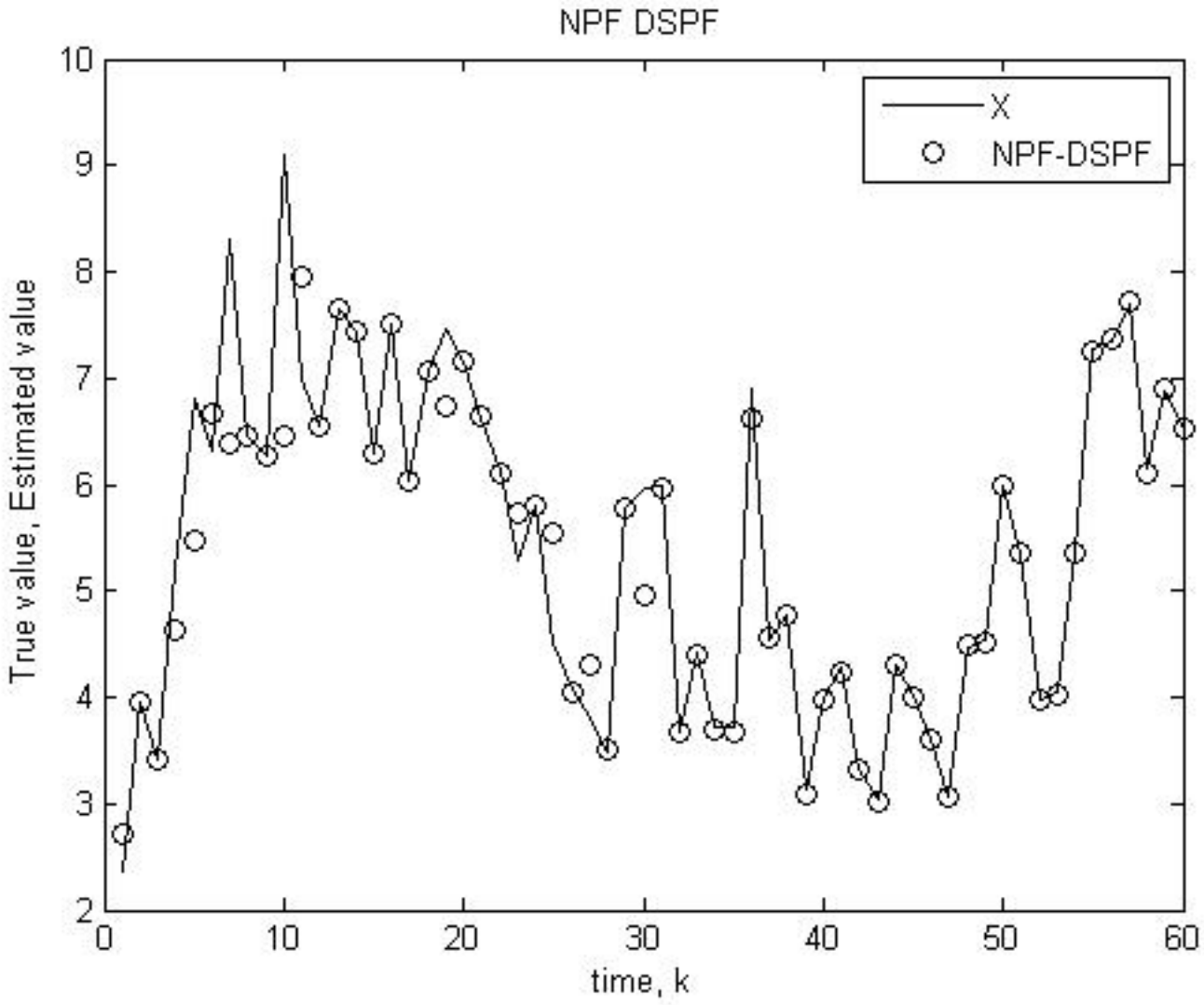
The rate constants are given by,

$$k = \begin{bmatrix} 0.5 & 0.05 & 0.2 & 0.01 \end{bmatrix}^T \tag{8.4}$$

The stoichiometry is given by,

$$\nu = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \end{bmatrix} \tag{8.5}$$

Figure 15: Nested Particle Filter- DSPF



The reaction rates are given by,

$$\gamma = \begin{bmatrix} k_1 c_A - k_2 c_B c_C \\ k_3 c_B^2 - k_4 c_C \end{bmatrix} \tag{8.6}$$

The state vector and the measurement equations are given as,

$$x = \begin{bmatrix} c_A & c_B & c_C \end{bmatrix}^T \tag{8.7}$$

$$y = \begin{bmatrix} RT & RT & RT \end{bmatrix} x \tag{8.8}$$

where c denotes the concentration of the species, R is the gas constant and T is the reactor temperature. It is assumed that ideal gas law holds good for this system.

The model for a well mixed, constant volume, isothermal batch reactor is

$$\dot{x} = f(x) = \nu^T \gamma \tag{8.9}$$

$$x_0 = \begin{bmatrix} 0.5 & 0.05 & 0 \end{bmatrix}^T \tag{8.10}$$

The state of this system is estimated using the following parameters which are also the initial conditions:

$$\Delta t = t_{k+1} - t_k = 0.25 \tag{8.11}$$

$$\Pi_0 = \text{diag}(0.5^2, 0.5^2, 0.5^2) \tag{8.12}$$

$$G_k = \text{diag}(1, 1, 1) \tag{8.13}$$

$$Q_k = \text{diag}(0.001^2, 0.001^2, 0.001^2) \tag{8.14}$$

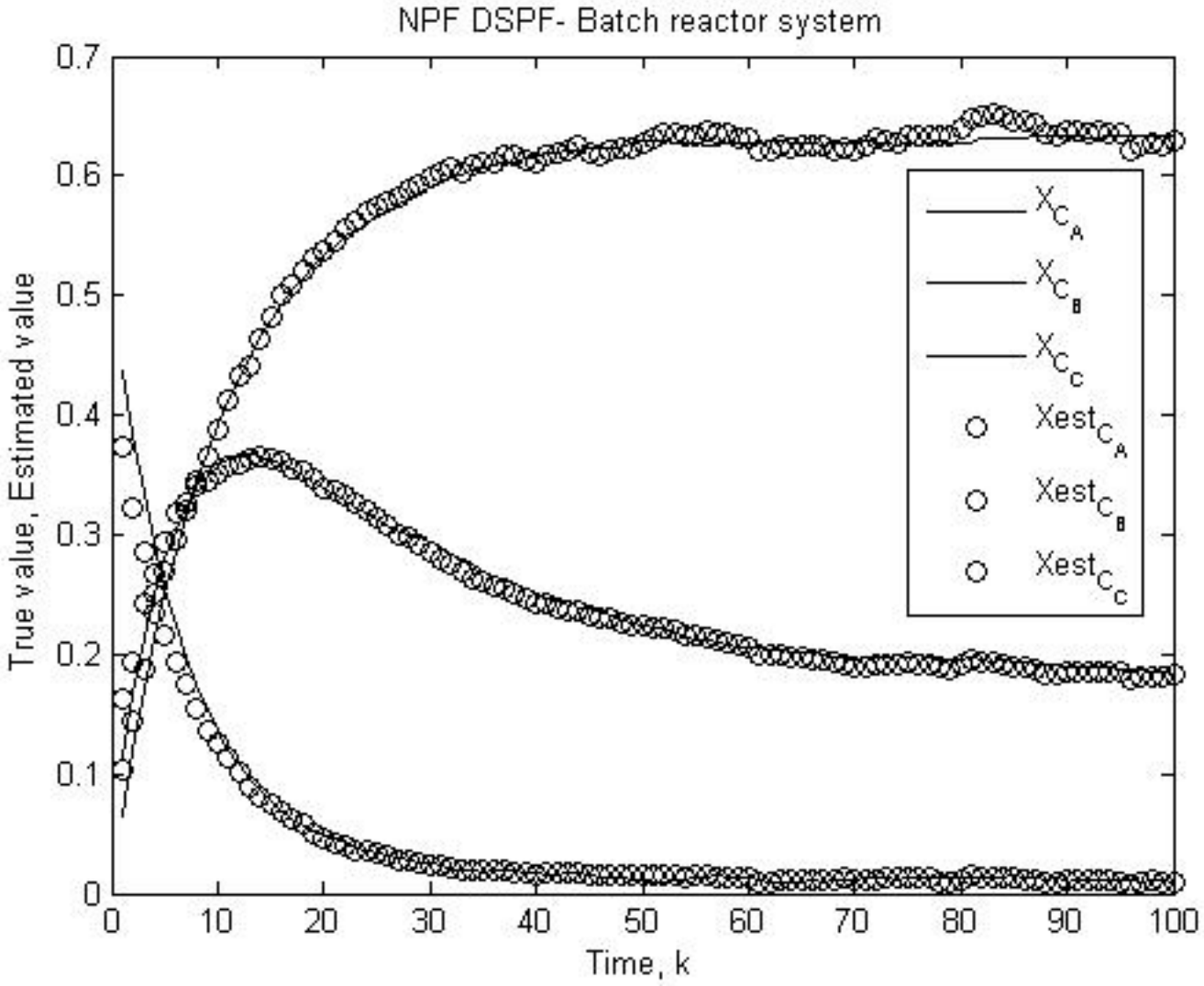
$$R_k = 0.25^2 \tag{8.15}$$

$$\bar{x} = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix}^T \tag{8.16}$$

where, Δt is the time step between two time instances, $\bar{x}$ is the initial conditions for the filter, Q_k and R_k are the state and the measurement noise covariances. The aim of this example is to provide consolidation to the fact that nested particle filter with direct sampling particle filter as its update equation shows good results when compared with any other filter.

The Fig. 16 illustrates how the nested particles filter performs when direct sampling particle filter is the update equation.

Figure 16: NPF- DSPF Batch reactor



CHAPTER IX

CONCLUSIONS

The main aim of the thesis was to obtain an algorithm for the nested particles filter, which works on the concept of local linearization. Several topics were discussed in detail in order to achieve best understanding of estimation techniques such as, unscented transform, Monte-Carlo method, importance sampling, resampling, and local linearization.

Direct sampling particle filter, ensemble Kalman filter and unscented particle filter were discussed in detail. An example that describes each of the system and a comparisons among estimation techniques was also provided.

The nested particles filter was analyzed for the possibilities of using an ensemble Kalman filter and/or the direct sampling particle filter as an update method and the advantages of one technique over the other were explained.

An examplewas performed where all nonlinear filters were analyzed and compared with respect to that scenario. A detailed comparison of why an unscented Kalman filter is better than an extended Kalman filter was made in order to understand the approximations made.

The choice of the importance density and the significance of choosing the transition prior as the proposal density were explained in detail. This lays the foundation for the development of the nested particles filter which, in turn, uses the DSPF and EnKF along with the concept of local linearization.

Finally, the algorithm for the nested particle filter with respect to both the update methods was explained, and a comparison was made. Even though each of the filters showed to have their own disadvantages, the direct sampling particle filter appears as the one with more accurate results. The reason for this performance can be attributed to, the ensemble Kalman filter, still using the Gaussian assumption as in the case of an Kalman filter. Whereas, the direct sampling particle filter is termed to perform on a constrianed state space model which does not involve any Gaussian assumptions or the approximations made in the other nonlinear filters.

BIBLIOGRAPHY

[1] Arulampalam, M., Maskell, S., Gordon, N., Clapp, T., A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Trans. on Signal Processing.*, vol. 50, no. 2,174-188, Feb. 2002.

[2] Bolic, M., Djuric, P. M., Hong, S., Resampling Algorithms for Particle FIlters: A Computational Complexity Perspective, *EURASIP Journal on Applied Signal Processing*, vol. 15,2442-2450, 2004.

[3] Chen, T., Morris, J., Martin, E., Particle filters for state and parameter estimation in batch processes, *Journal of Process Control*, vol. 15, 665-673, 2005.

[4] Evensen, G., The Ensemble Kalman Filter: theoretical formulation and practical implementation, *Ocean Dynamics*, vol. 53, 343-367, May 2003.

[5] Gordon, N. J., Salmond, D. J., Smith, A. F. M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings*, vol. 140, no. 2, 107-113, Apr. 1993.

[6] Haseltine, E. L., Rawlings, J. B., Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation, *Ind. Eng. Chem. Res.*, vol. 44, no. 8,2451-2460, 2005.

[7] Julier, S. J., Uhlmann, J. K., New extension of the Kalman filter to nonlinear systems, *Proc. SPIE*, vol. 3068, no. 182, 1997.

[8] Julier, S.J., Uhlmann, J.K., Unscented Filtering and Nonlinear Estimation, *Proceedings of the IEEE.*, vol.92, no. 3, 401-422, Mar. 2004.

[9] Kandepu, R., Foss, B., Imsland, L., Applying the unscented Kalman filter for nonlinear state estimation, *Journal of Process Control*, vol. 18, 753-768, 2008.

[10] Liu, J. S., Chen, R., Sequential Monte Carlo Methods for Dynamic Systems, *Journal of the American Statistical Association*, vol. 93, no. 443, 1032-1044, Sep. 1998.

[11] Oppenheim, G., Philippe, A., de Rigal, J., The particle filters and their applications, *Chemometrics and Intelligent Laboratory Systems*, vol. 91, 87-93, 2008.

[12] Papoulis, A., *Probability, random variables, and stochastic processes* McGraw-Hill, 1984.

[13] Rawlings, J. B., Bakshi, B. R., Particle filtering and moving horizon estimation, *Computers and Chemical Engineering*, vol. 30, 1529-1541, 2006.

[14] Salahshoor, S., Bayat, M. R., Mosallaei, M., 'Design of Instrumentation Sensor Networks for Non-linear Dynamic Processes using Extended Kalman Filter, *Iran. J. Chem. Chem. Eng.*, vol. 27, no. 3, 2008.

[15] Simon, D., *Optimal State Estimation: Kalman, H-Infinity and Nonlinear Approaches.* John Wiley and Sons, 2006.

[16] Ungarala, S., A direct sampling particle filter from approximate conditional density function supported on constrained state space, *Computers and Chemical Engineering*, vol. 35, no. 6, 1110-1118, Jun. 2011.

[17] Ungarala, S., Dolence, E., Li, K., Constrained extended Kalman filter for nonlinear state estimation, *Dynamics and Control Process Systems*, vol. 8, no. 1, 63-68, 2007.

[18] Ungarala, S., Sequential Monte Carlo Filtering Using Nested Particles with Local Gaussian Assumptions, Advanced Control of Chemical Processes, vol. 8, no. 1, 2012.

[19] van der Merwe, R., Doucet, A., de Freitas, N., Wan, E., The Unscented Particle Filter, *Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department*, 349-354, Aug. 2000.

[20] Wan, E. A., Van der Merwe, R., The unscented Kalman filter for nonlinear estimation, *Proc. Symp. Adaptive Syst. Signal Process., Commun. Contr.*, 153-158, Oct. 2000.

[21] Stratonovich, R.L., Conditional Markov Processes, *Theor. Probability Appl.* vol. 5, no. 2, 156-178, 1960.

[22] Jazwinski, A. H., Filtering for Nonlinear Dynamical Systems, *IEEE Trans. Automatic Control*, vol. 11, 765-766, 1966.

[23] Kalman, R. E., Bucy, R. S., A New Approach to Linear Filtering and Prediction Problems, *Trans. ASME, Ser. D: J. Basic Eng.*, vol. 82, 35-45, Mar. 1960.

[24] Kushner, H. J., On the Dynamical Equations of Conditional Probability Density Functions, with Applications to Optimal Stochastic Control Theory, *J. Math. Anal. Appl.* vol. 8, 332-344, 1964.

[25] Wonham, W. M., Stochastic Problems in Optimal Control, *IEEE Intern. Conv. Record*, vol. 11, 1963.

APPENDIX

1 Matlab files for nonlinear filter

Code for the three filters- PF, DSPF, EnKF

```
MSE_part= [];
MSE_dspf = [];
MSE_enkf = [];
%MSE_ekf = [];
for count = 1:100
% SIMULATION
% Initial state
x =1;
n = length(x);
m = 1;
% Noise covariances
A = 3;
B = 0.5;
R = 0.00001;
% Number of time steps
ts = 60;
t = [1:ts];
k = 1;
Y = zeros(size(R,1),ts);
X = zeros(size(x,1),ts);
x_function = @tseries_x;
X(:,1) = feval(x_function, x,1) + gamrnd(A,B,1);
% Non-linear equations
for k = 2:ts
```

```
X(:,k) = feval(x_function, X(:,k-1),k) +gamrnd(A,B,1);
end
for k = 1:ts
    if k <=30
        y_function = @tseries1_y;
    else
        y_function = @tseries2_y;
    end
Y(k) = feval(y_function,X(:,k))+ R^0.5*randn;
end
[X_part,CPUT_part ] = particlefilter(X, Y, k,A, B, R );
 mse_part = sum((X-X_part).^2)/ts;
 MSE_part = [MSE_part; mse_part];
 [X_dspf,CPUT_dspf ] = dspf(X, Y, k,A, B, R );
 mse_dspf = sum((X-X_dspf).^2)/ts;
 MSE_dspf = [MSE_dspf; mse_dspf];
 [X_enkf,CPUT_enkf ] = enkf(X, Y, k,A, B, R );
 mse_enkf = sum((X-X_enkf).^2)/ts;
 MSE_enkf = [MSE_enkf; mse_enkf];
end
Particle = [mean(MSE_part'), std(MSE_part'),sqrt(mean(MSE_part)),CPUT_part]
DSPF = [mean(MSE_dspf'), std(MSE_dspf'),sqrt(mean(MSE_dspf)),CPUT_dspf]
ENKF = [mean(MSE_enkf'), std(MSE_enkf'),sqrt(mean(MSE_enkf)),CPUT_enkf]
plot (t,X,t,X_part,'x',t,X_dspf,':',t,X_enkf,'o');
xlabel('time, k');
ylabel('true value, estimated value');
```

```
title('Tseries system-Particle Filter, DSPF, EnKF');
legend('X', 'Estimate- PF','Estimate- DSPF', 'Estimate-EnKF');
```

%tseries_x.m

```
function x_out = tseries_x(x,k)
n = k(1);
x_out = 1+ sin(pi*(n-1)/25) + 0.5*x;
```

%tseries1_y.m

```
function y_out = tseries1_y(x)
y_out = 0.5*x.^2;
```

%tseries2_y.m

```
function y_out = tseries2_y(x)
y_out = 0.5*x - 2*ones(size(x));
```

%particlefilter.m

```
% ESTIMATION
% Particle Filter
function [X_est,CPUT ] = particlefilter(X, Y, k,A, B, R )
xp = 1;
P = 1;
S = 200;
ts = 60;
t = [1:ts];
x_function = @tseries_x;
XP = xp + (P^0.5)*randn(1,S);
```

```
X_est = zeros(size(xp,1),size(Y,2));

P_est= zeros(size(xp,1),size(Y,2));

XP_particles =[];

tt = clock;

for k = 1:ts

    if k <=30

        y_function = @tseries1_y;

    else

        y_function = @tseries2_y;

    end

XP = feval(x_function,XP,k)+ gamrnd(A,B,1,S);

YP = feval(y_function, XP)+ R^0.5*randn ;

% weights and normalized weights

w = 1/sqrt(2*pi*R) * exp(-(Y(k)-YP).^2 / (2 * R));

w = w/sum(w);

%resampling

[Index]= resampleResidual(w);

XP = XP(Index);

XP_particles(:,k) = XP;

xp = mean(XP);

P = cov(XP);

% estimated x

X_est(:,k) = xp';

P_est(:,k)= P;

end

CPUT = etime(clock,tt);
```

%**dspf.m**

```
%direct sampling particle filter
function [X_est,CPUT ] = dspf(X, Y, k,A, B, R )
xp = 1;
P = 1;
S = 200;
ts = 60;
t = [1:ts];
XP = xp + (P^0.5)*randn(1,S);
x_function = @tseries_x;
X_est = zeros(size(xp,1),size(Y,2));
P_est= zeros(size(xp,1),size(Y,2));
XP_particles =[];
tt = clock;
for k = 1:ts
    if k <=30
        y_function = @tseries1_y;
    else
        y_function = @tseries2_y;
    end
XP = feval(x_function,XP,k)+ gamrnd(A,B,1,S);
YP = feval(y_function, XP)+ R^0.5*randn ;
    mu =mean(XP');
    c = cov(XP');
    sigma = (diag(c)^0.5);
    a = mu - 3*sigma;
```

```
    b = mu + 3*sigma;
    XP = a+(b-a)*rand(1,S);
    Pk = sigma.^2;
W = exp(-0.5*(((XP-mu).^2/sigma^2) + (Y(k)-feval(y_function, XP)).^2/R));
    W = W/sum(W);
    [Index]= resampleResidual(W);
    XP = XP(:,Index);
xp = mean(XP);
P = cov(XP);
% estimated x
X_est(:,k) = xp';
P_est(:,k)= P;
end
 CPUT = etime(clock,tt);
```

%enkf.m

```
% ensemble kalman filter
function [X_est,CPUT ] = enkf(X, Y, k,A, B, R )
xp = 1;
P = 1;
S = 200;
ts = 60;
t = [1:ts];
XP = xp + (P^0.5)*randn(1,S);
X_est = [];
% prior mean and covariance
tt = clock;
```

```
x_function = @tseries_x;
for k = 1:ts
    if k <=30
        y_function = @tseries1_y;
    else
        y_function = @tseries2_y;
    end
XP = feval(x_function,XP,k)+ gamrnd(A,B,1,S);
YP = feval(y_function, XP)+ R^0.5*randn ;
    SAMP =[XP',YP'];
    covariance = COV(SAMP);
   Pxx = covariance(1,1);
   Pyy = covariance(2,2);
   Pxy = covariance(1,2);
    K = Pxy'*inv(Pyy);
    XP = XP+ K*(Y(k) - YP);
    X_est = [X_est,MEAN(XP')];
end
CPUT = etime(clock,tt);
```

%resampleResidual.m

```
function [ indx ] = resampleResidual( w )
M = length(w);
% "Repetition counts" (plus the random part, later on):
Ns = floor(M .* w);
% The "remainder" or "residual" count:
R = sum( Ns );
```

```
% The number of particles which will be drawn stocastically:
M_rdn = M-R;
% The modified weights:
Ws = (M .* w - floor(M .* w))/M_rdn;
% Draw the deterministic part:
i=1;
for j=1:M,
    for k=1:Ns(j),
        indx(i)=j;
        i = i +1;
    end
end;
% And now draw the stocastic (Multinomial) part:
Q = cumsum(Ws);
Q(M)=1; % Just in case...
while (i<=M),
    sampl = rand(1,1);  % (0,1]
    j=1;
    while (Q(j)<sampl),
        j=j+1;
    end;
    indx(i)=j;
    i=i+1;
end
```

2 Nested particles filter matlab files

Comparison between EnKF and DSPF update

```
% DSPF update and ENKF update comparison

clc;

clear all;

MSE_enkf = [];

RMSE_enkf =[];

STDN_enkf = [];

MSE_dspf = [];

RMSE_dspf =[];

STDN_dspf = [];

for count= 1:100

% SIMULATION

% Initial state

x =1;

n = length(x);

m = 1;

% Noise covariances

A = 3;

B = 0.5;

R = 0.01;

% Number of time steps

ts = 60;

t = [1:ts];

k = 1;

Y = zeros(size(R,1),ts);
```

```
X = zeros(size(x,1),ts);
x_function = @tseries_x;
X(:,1) = feval(x_function, x,0) + gamrnd(A,B,1);
for k = 2:ts
% Non-linear equations
X(:,k) = feval(x_function, X(:,k-1),k) + gamrnd(A,B,1);
end
for k = 1:ts
    if k <=30
        y_function = @tseries1_y;
    else
        y_function = @tseries2_y;
    end
Y(k) = feval(y_function,X(:,k))+ R^0.5*randn;
end
N = 50;
M = 500;
[X_enkf, CPUT_enkf] = npf_enkf(X,Y, A, B,R,ts,k,N,M);
mse_enkf = sum((X-X_enkf).^2)/(ts*n);
rmse_enkf = sqrt(mse_enkf);
MSE_enkf = [MSE_enkf,mse_enkf'];
RMSE_enkf = [RMSE_enkf,rmse_enkf'];
[X_dspf, CPUT_dspf] = npf_dspf(X,Y, A, B,R,ts,k,N,M);
mse_dspf = sum((X-X_dspf).^2)/(ts*n);
rmse_dspf = sqrt(mse_dspf);
MSE_dspf = [MSE_dspf,mse_dspf'];
```

```
RMSE_dspf = [RMSE_dspf,rmse_dspf'];
end
NPF_ENKF = [mean(MSE_enkf'),mean(RMSE_enkf'),std(MSE_enkf'),CPUT_enkf]
NPF_DSPF = [mean(MSE_dspf'),mean(RMSE_dspf'),std(MSE_dspf'),CPUT_dspf]
```

%npf_enkf.m

```
% NPF_EnKF_Gaussian proposal
function [X_est,CPUT] = npf_enkf(X,Y, A, B,R,ts,k,N,M)
xp = 1;
P = 1;
x_function = @tseries_x;
X_est = zeros(size(xp,1),size(Y,2));
P_est= zeros(size(xp,1),size(Y,2));
XP = xp + (P^0.5)*randn(1,N);
% xprop = XP;
% pprop = P*XP.^0;
XN = [];
XN = repmat(XP, M,1) + P^0.5*randn(M,N);
tt = clock;
for k = 1:ts
   if k <= 30
       y_function = @tseries1_y;
   else
       y_function = @tseries2_y;
   end
   XN = feval(x_function,XN,k)+ gamrnd(A,B,M,N);
   YN = feval(y_function,XN) + R^0.5*randn(M,N);
```

```
   SAMP = cov([XN YN]);
   Pxy = diag(SAMP(1:N,N+1:2*N))';
   Pyy = diag(SAMP(N+1:2*N,N+1:2*N))';
   XN = XN + repmat(Pxy./Pyy,M,1).*(repmat(Y(:,k),M,N)-YN);
   xprop = mean(XN);
   pprop = var(XN);
   XP_new = xprop + (pprop.^0.5).*randn(1,N);
   dummy = XP_new - feval(x_function, XP,k);
   tp = (dummy.^(A-1).*exp(-dummy/B)/(A^B*gamma(A)))';
   lk = (exp(-(Y(:,k)-feval(y_function,XP_new)).^2./(2*R)))';
   prop = (exp(-(XP_new-xprop).^2./(2*pprop)))';
   w = lk.*tp./prop;
   w = w/sum(w);
   ind = resampleResidual(w);
   XP = XP_new(:,ind);
   xp = mean(XP');
   P = cov(XP');
   X_est(:,k) = xp';
   P_est(:,k) = P;
   XN = XN(:,ind);
   XN = XN - repmat(mean(XN),M,1) + repmat(XP,M,1);
end
CPUT = etime(clock,tt);
```

%npf_dspf.m

```
% NPF with an DSPF update
function [X_est,CPUT] = npf_dspf(X,Y, A, B,R,ts,k,N,M)
```

```
xp = 1;
P = 1;
x_function = @tseries_x;
X_est = zeros(size(xp,1),size(Y,2));
P_est = zeros(size(xp,1),size(Y,2));
XP = xp + (P^0.5)*randn(1,N);
% xprop = XP;
% pprop = P*XP.^0;
XN = [];
Ind = [];
XN = repmat(XP, M,1) + P^0.5*randn(M,N);
tt = clock;
for k = 1:ts
   if k <= 30
       y_function = @tseries1_y;
   else
       y_function = @tseries2_y;
   end
   XN = feval(x_function,XN,k)+ gamrnd(A,B,M,N);
   YN = feval(y_function,XN) + R^0.5*randn(M,N);
   mu = mean(XN);
   c = var(XN);
   sigma = sqrt(c);
   a = mu-3*sigma;
   b = mu+ 3*sigma;
   for loop = 1:N
```

```
    XN(:,loop) = a(:,loop)+(b(:,loop)-a(:,loop))*rand(M,1);
    w(:,loop) = exp(-0.5*(((XN(:,loop)-mu(:,loop)).^2)./sigma(:,loop).^2 ...
        + ((Y(k) - feval(y_function,XN(:,loop))).^2*R^(-1))));
    w(:,loop) = w(:,loop)/sum(w(:,loop));
    [Index] = resampleResidual(w(:,loop));
    Ind(:,loop) = Index';
    end
    % update
    XN = XN(Ind);
    % Importance density
    xprop = mean(XN);
    pprop = var(XN);
    XP_new = xprop + (pprop.^0.5).*randn(1,N);
% weights
    dummy = XP_new - feval(x_function, XP,k);
    tp = (dummy.^(A-1).*exp(-dummy/B)/(A^B*gamma(A)))';
    lk = (exp(-(Y(:,k)-feval(y_function,XP_new)).^2./(2*R)))';
    prop = (exp(-(XP_new-xprop).^2./(2*pprop)))';
    W = lk.*tp./prop;
    W = W/sum(W);
     ind = resampleResidual(W);
    XP = XP_new(:,ind);
    xp = mean(XP');
    P = cov(XP');
    X_est(:,k) = xp';
    P_est(:,k) = P;
```

```
   XN = XN(:,ind);
  XN = XN - repmat(mean(XN),M,1) + repmat(XP,M,1);
end
CPUT = etime(clock,tt);
```