SEQUENTIAL MONTE CARLO METHODS WITH APPLICATIONS TO COMMUNICATION CHANNELS

A Thesis

by

SIRISH BODDIKURAPATI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2009

Major Subject: Electrical Engineering

SEQUENTIAL MONTE CARLO METHODS WITH APPLICATIONS TO COMMUNICATION CHANNELS

A Thesis

by

SIRISH BODDIKURAPATI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Henry Pfister
Committee Members,	Jean-Francois Chamberland
	Gregory Huff
	Radu Stoleru
Head of Department,	Costas Georghiades

December 2009

Major Subject: Electrical Engineering

ABSTRACT

Sequential Monte Carlo Methods with Applications to Communication Channels.

(December 2009)

Sirish Boddikurapati, B.E., Andhra University

Chair of Advisory Committee: Dr. Henry Pfister

Estimating the state of a system from noisy measurements is a problem which arises in a variety of scientific and industrial areas, including signal processing, communications, statistics and econometrics. Recursive filtering is one way to achieve this by incorporating noisy observations as they become available with prior knowledge of the system model.

Bayesian methods provide a general framework for dynamic state estimation problems. The central idea behind this recursive Bayesian estimation is computing the probability density function of the state vector of the system conditioned on the measurements. However, the optimal solution to this problem is often intractable because it requires high-dimensional integration. There are many new methods of filtering for the general case. The main emphasis of this thesis is on one such recently developed filter, the particle filter.

A detailed introduction to particle filters is provided, as well as applications to various communication channels. This thesis provides a particle filtering method for calculating the capacity of wireless channels and the calculation of information rates over optical fibers. Important conclusions are drawn for using particle filtering methods in fiber optic channels.

To My Loving Family and Friends

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and deepest thanks to my advisor, Dr. Henry Pfister. I thank him for his excellent instruction, advice, continuous motivation, encouragement and supervision of my thesis.

I would also like to thank my graduate committee members, Dr. Jean-Francois Chamberland, Dr. Radu Stoleru and Dr. Gregory Huff for their constant monitoring and help they offered at every moment of my thesis work.

Most of all, I would like to thank my family to whom I dedicate this work, especially my parents for their graceful support.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

A. Overview

In many application areas including signal processing, statistics, communications, and econometrics, the observations of dynamic system are used to analyse the system [1,2]. The analysis may be based on the values of the observations or on an underlying state which is related to the observations made. Applications for the former include monitoring of rainfall or the share values of companies. In this case, the past values of observations will be studied in order to learn and predict the future values. Applications for the latter include tracking of a target moving in a two dimensional space, when the observations are the bearings of the target at different times, corrupted by some random noise. Here the underlying state can be either position or velocity of the target. There are many approaches to this problem but a sequential analysis is favored. In this approach, the current estimate of the state of the system is updated based on the previous state estimate and the current measurement.

There are two major advantages with this approach. First, the computational cost is much less because the analysis depends only on the previous state instead of all past states. Secondly, the storage capacity required is much less. Only the current posterior state estimate needs to be stored. But, the disadvantage of such an approach will become apparent in cases where the posterior state estimate can not be exactly calculated. In such cases, the estimation error of the posterior state may get accumulated as the system evolves in time.

This thesis mainly deals with the analysis of dynamic systems with the underlying

---

The journal model is *IEEE Transactions on Automatic Control.*

states being continuous or discrete. In particular, the use of sequential Monte Carlo methods, which is a recently developed technique for use in filter theory, is studied [2,3,4]. A new class of filters, called the Particle filter, are thus developed. The rest of this chapter will discuss the Bayesian approach to filtering theory and the extension to Sequential/Recursive Bayesian Estimation method that is the basis for a Particle filter. The final section will give a brief outline of the rest of the thesis.

B. Bayesian Inference for Non-Linear Filtering

Bayesian methods form a rigorous general frame work for dynamic state estimation. In the Bayesian framework, all the unknown quantities are treated as random variables. *A priori* knowledge of the system is often available for the formulation of Bayesian models. The basic approach of this method is to construct a posterior probability density of the state based on all available information. Using Bayes' theorem, a posterior density can be computed from the prior distributions and the likelihood function [5]. Inference of unknown quantities and their related statistics are made based on the resultant posterior density. In reality, however, observations usually occur sequentially in time and estimation of the unknown values is often required on-line. This motivates the idea of updating the posterior distribution as the observation data becomes available. Storing all the observational data may not be necessary if the posterior distribution is updated sequentially in time. In recursive Bayesian estimation [3], optimal solution is calculated from the a posterior density based on certain cost function.

In linear systems with Gaussian process and measurement noise, an optimal closed-form solution is the well-known *Kalman filter* [2,6]. However, in the case of nonlinear or non-Gaussian problems, a closed form solution to problem is often

intractable since it requires high dimensional integration. Therefore, approximate non-linear filters [1,3] have been proposed. The most common approach is *Extended Kalman filter (EKF)* [2,6], which approximates the model by a linearized version and then use the optimal Kalman filter with this approximate model. The linearization is done by using the Taylor series expansion for the non-linear terms. This filter works well for weakly non-linear systems as the higher order terms in the expansion can be considered negligible and the linear system effectively models the actual non-linear system. But for high degree of non-linearity, the higher order terms can not be considered negligible and this results in additional computational complexity [2,6]. Moreover, the EKF assumes Gaussian properties for the noise which is not always the case with the real systems. There are many practical applications with non-linear and non-Gaussian problems namely, localization of robots, estimating noisy digital communications signals, image processing, and aircraft tracking using radar measurements [1,3]. Numerical integration [1,3] is another approach that could be used in non-linear, non-Gaussian cases but it is usually too computationally expensive to be used in practical applications. Thus, the idea of Monte Carlo simulation based filters came into being.

Although the idea of Monte Carlo simulation [3] originated in the late 1940s, its popularity in the field of filtering started in 1993 starting with the algorithm proposed by Gordon et al. [4]. The Monte Carlo technique [4,7,8] is a kind of stochastic sampling approach aiming to tackle the complex systems which are analytically intractable. The power of Monte Carlo methods is that they can approximate the solutions of difficult numerical integration problems [4]. These methods fall into two categories, namely, *Markov chain Monte Carlo (MCMC)* methods for batch signal processing and *Sequential Monte Carlo (SMC)* methods for adaptive signal processing.

The Sequential Monte Carlo [7,8,9,10] approaches have attracted more and more attention in different areas with many applications in signal processing, statistics, machine learning, econometrics, automatic control, tracking, communications, biology and many others. One of the attractive merits of these approaches lies in the fact that they allow on-line estimation by combining the powerful Monte Carlo sampling methods with Bayesian inference at an expense of reasonable computational cost [7,8]. The sequential Monte Carlo approach has been used in parameter estimation and state estimation. This SMC approach is known variously as *Particle Filtering*, *Boot-strap filtering*, *Condensation Algorithm*, and in more subtle words as, the *Survival of the Fittest* [2,3,7,8,11,12,13].

In comparison with standard approximation methods, such as the EKF, the advantage of *Particle filtering* is that it does not involve linearizations around current estimates. Instead, the representation of the desired distributions is approximated by discrete random measures [7,8]. The basic idea of particle filter is to use a number of independent random variables, called *particles*, that are sampled directly from the state space to represent the posterior probability, and to update the posterior based on the new observations; the particle system is properly located, weighted, and propagated recursively according to the Bayesian rule [1,2,3,10]. Particle filtering methods are not limited by nonlinearity and non-Gaussianity constraints and therefore have the potential to push filtering theory beyond these challenges for many real-time systems.

C. Thesis Outline

The main emphasis of the research is based on studying the particle filtering approach to estimate the information sent over different communication channels including

fading channels (flat), adaptive channels, and optical channels. A brief outline of the subsequent chapters in this thesis is given:

Chapter II introduces the recursive Bayesian approach for estimating the state of a system. The state-space model that will be used in this research is presented. It also gives a little introduction to the different Monte Carlo methods. It also reviews the different filtering techniques starting from the one proposed by Kalman and Bucy in 1961.

Chapter III introduces the Particle filter. A detailed derivation of sequential importance sampling (SIS), which is the basis for the particle filtering technique is presented. The degeneracy phenomenon, resampling and choice of sampling density in particle filter are emphasized. An example of particle filtering approach to a simple non-linear system, taken from [2], is dealt and results are studied.

Chapter IV introduces to the application of particle filter in detecting a signal in noisy environment through a flat faded wireless channel. The first few sections of the chapter deals with the channel model. Then the general procedure of particle filtering is discussed. The final section shows some simulation results to help understand the application better.

Chapter V discusses the calculation of Information Rates using a particle filtering approach. It describes the method used by Dauwels and Loeliger in [14] and extends the theory to be applied to the adaptive channel that is introduced in the previous chapter.

Chapter VI introduces the usage of filters for problems involving Stochastic Differential Equations (SDEs). These kind of non-linear continuous-discrete filtering problems have using Taylor series expansion based approximation methods for filtering purposes, however the errors in approximations lead to limitations in some real time applications. The recent discrete-time filtering algorithms, the unscented

Kalman filter and particle filters, are introduced here for use in continuous-discrete filtering problems by using the an example of tracking angular velocity of a simple pendulum, a problem dealt previously by Simo Sarkka [15] and others.

Chapter VII introduces the propagation phenomenon through an optical fiber. One of the method to calculate the capacity of the optical channel as discussed by Essiambre and others in [16] is introduced for both QAM and ring constellations with a root raised cosine modulation scheme.

Chapter VIII presents some conclusions, and highlights some areas of possible future research in this field of filter theory.

CHAPTER II

BAYESIAN FILTERING

Bayesian filtering is a branch of probability theory that models the uncertainty in the world (e.g., the outcomes of interest) by combining prior knowledge and observational evidence. Bayesian analysis, interpreting the probability as a conditional measure, is one of the popular methods in many cases. However, for many problems in communications and signal processing, an estimate is required every time a measurement is received. In these cases, a recursive filter is a convenient solution. A recursive filtering approach means that received data is processed sequentially rather than as a batch so that it is not necessary to store the complete data set or to reprocess existing data if a new measurement becomes available.

In this chapter, an overview of the recursive Bayesian approach is provided. The general mathematical formulation of the state-space model that will be studied in this thesis will be presented next. Further, the optimal filtering technique, Kalman filter, for a linear system in the presence of Gaussian noise is summarized. The concept of Monte Carlo sampling for solving the intractable integrals is discussed. A detailed derivation of sequential importance sampling (SIS) which is the basis for the particle filtering technique is presented further in next chapter.

In Bayesian reference, all the uncertainties (including states, parameters, which are either time-varying or fixed) are treated as random variables. The inference is performed with in the Bayesian framework given all the available information. The objective of Bayesian inference is to use the priors and causal knowledge, quantitatively and qualitatively, to infer the conditional probability, given a finite number of observations related to the state of the system. There are usually three levels of the probabilistic reasoning in Bayesian analysis. Starting with selecting a model given the

data and assumed priors; estimating the parameters to fit the data given the model and priors; and finally updating the parameters of the prior.

There are three types of intractable problems inherently related to the evaluation of *a posteriori density* $p(x|y)$ as given in [1].

- **Normalization**: Given the prior $p(x)$ and likelihood $p(y|x)$, the posterior $p(x|y)$ is obtained by the product of prior and likelihood divided by a normalizing factor. The expression for the posterior $p(x|y)$ is given by

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)}dx. \tag{2.1}$$

- **Marginalization**: Given the posterior $p(x, z|y)$, where z is a variable that is not of interest, the marginal posterior $p(x|y)$ is calculated by

$$p(x|y) = \int p(x, z|y)dz. \tag{2.2}$$

- **Expectation**: Given the conditional pdf $p(x|y)$, the expectation of the function $f(x)$ can be calculated as

$$E_{p(x|y)}[f(x)] = \int f(x)p(x|y)dx. \tag{2.3}$$

The recursive filtering approach means that received data is processed sequentially so that it is not necessary to store the complete data set. A state space model [2,6] must be used in such situations which is essentially a notational convenience used for estimation and control problems. The recursive bayesian approach can be better understood after introducing the state space model for such a problem.

A. State Space Form

The state of the system generally refers to the dynamic variables such as position, velocities and accelerations, which describe the physical state of the system. The noise in the measurements means that the measurements are uncertain. This means that even if the true state is known, the measurements will not be of deterministic form in terms of the state. The time evolution of the state is modeled as a dynamic system, which has a certain process noise. This noise is used for accounting the uncertainties that are present in the system dynamics.

The following generalised state space form through out this thesis. A state x of a dynamic system is assumed to change with time. To estimate the state of the system, measurements are made on the system at discrete intervals of time $\{t_1,t_2,...\}$. The state at time t_k is denoted by x_k and the corresponding measurement being denoted by z_k. The system model for such a system is given by

$$X_k = f(X_{k-1}, V_k). \tag{2.4}$$

where f : $\mathbb{R}^{n_X} \times \mathbb{R}^{n_V} \rightarrow \mathbb{R}^{n_X}$ is the system evolution function and $\{V_k\}$ represents a set of random variables with a known distribution. This set accounts for the process noise that is assumed to be in the system dynamics. Here n_X and n_V are the dimensions of the state of the system and process noise respectively. At time t_0, i.e. prior to any measurement being taken, the state is assumed to be having a prior density of $p(x_0)$.

The measurement z_k at time t_k, drawn from the random variable Z_k, is given by the measurement model

$$Z_k = h(X_k, N_k). \tag{2.5}$$

where $h : \mathbb{R}^{n_X} \times \mathbb{R}^{n_N} \rightarrow \mathbb{R}^{n_Z}$ is the system evolution function and $\{n_k\}$ represents the realization of noise in the measurements taken on the system. Here n_X and n_N

are the dimensions of the state of the system and measurement noise respectively. Also the sequences $\{V_k\}$ and $\{N_k\}$ are assumed to be independent of each other. The functions f and h are assumed to be known and so are the distributions for the noise terms explained above.

B. Recursive Bayesian Solution

The Bayesian approach to infer the information about the state, $\{X_k\}$ at time t_k, is to calculate the posterior density conditional on the measurements available for the system. If $Z_k = \{z_1, z_2, z_3, ..., z_k\}$ is the set of the available measurements till time k, then the Bayesian solution is to find the posterior density of $p(x_k|Z_k)$. This density accounts for all the information that can learnt about the state of the system given all the measurements until that point of time. Also, if this density is known, then the optimal estimates of the state can be obtained. In other words, the estimate of x_k conditioned on the measurements Y_k is given by the conditional expectation of x_k with Y_k

$$E(x_k|Y_k) = \int x_k p(x_k|Z_k) dx_k. \tag{2.6}$$

This can be extended to estimating functions of the state instead of the state itself, i.e., if $g(.)$ is a function of the state that needs to be analyzed, then we can use the posterior density to calculate the optimal estimate of $g(x_k)$ as

$$E(g(x_k)|Y_k) = \int g(x_k) p(x_k|Z_k) dx_k. \tag{2.7}$$

The key to calculating the posterior density for the state is Bayes' theorem [14], which states that

$$p(x|z) \propto p(x)p(z|x). \tag{2.8}$$

This means that the value of the posterior density for x conditioned on the

measurement z is proportional to the prior value of x multiplied by the likelihood function for the measurement given the state values. But the direct calculation of such a posterior density has its own difficulties as it includes high-dimensional integration and this will in turn result in a drastic increase in the computational cost. This can be overcome by using a sequential scheme, called the recursive Bayesian filter [1]. The formulation of such a filter includes two stages, namely, the *prediction* stage and the *update* stage. Two assumptions are used to derive the recursive Bayesian filter as shown in [1]:

- The states follow a first-order Markov process i.e.,

$$p(x_k|x_{0:k-1}) = p(x_k|x_{k-1}) \tag{2.9}$$

- The observation is conditionally independent of the given states, i.e.,

$$p(z_k|x_{0:k}) = p(z_k|x_k). \tag{2.10}$$

1. Prediction Stage

The prediction stage uses the system model to predict the density function of the state forward from one measurement time to the next. Since the state is usually subject to unknown disturbances (modeled as random noise), the prediction generally translates, deforms, and spreads the state density function. Specifically, given the value of $p(x_{k-1}|z_{1:k-1})$, which is already available at time t_{k-1}, this stage involves the calculation of the pdf $p(x_k|z_{1:k-1})$. A detailed derivation for the state pdf prediction

is given below:

$$\begin{aligned} p(x_k|z_{1:k-1}) &= \frac{p(x_k, z_{1:k-1})}{p(z_{1:k-1})} \\ &= \frac{\int_{-\infty}^{\infty} p(x_k, x_{k-1}, z_{1:k-1}) dx_{k-1}}{p(z_{1:k-1})} \\ &= \frac{\int_{-\infty}^{\infty} p(x_k|x_{k-1}, z_{1:k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1}}{p(z_{1:k-1})} \\ &= \int_{-\infty}^{\infty} p(x_k|x_{k-1}, z_{1:k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1} \\ &= \int_{-\infty}^{\infty} p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1} \end{aligned} \tag{2.11}$$

The equation (2.11) is known as the *Chapman-Kolmogorov* (CK) equation [3]. It is to be noted that the above derivation makes use of the assumptions made on the Markovian states Eq (2.9) and the measurements Eq (2.10).

2. Update/Filter Stage

The update stage, as the name suggests, involves the update of the prediction density (2.11) based on the latest measurement available at that time. Specifically, given the measurement z_k at time k, the update stage uses the following few steps to compute the density $p(x_k|z_{1:k})$ via Bayes' rule [3].

$$\begin{aligned} p(x_k|z_{1:k}) &= \frac{p(x_k, z_{1:k})}{p(z_{1:k})} \\ &= \frac{p(z_k|x_k, z_{1:k-1}) p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1}) p(z_{1:k-1})} \\ &= \frac{p(z_k|x_k) p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \end{aligned} \tag{2.12}$$

where, the normalizing constant is given by

$$p(z_k|z_{1:k-1}) = \int_{-\infty}^{\infty} p(z_k|x_k) p(x_k|z_{1:k-1}) dx_k. \tag{2.13}$$

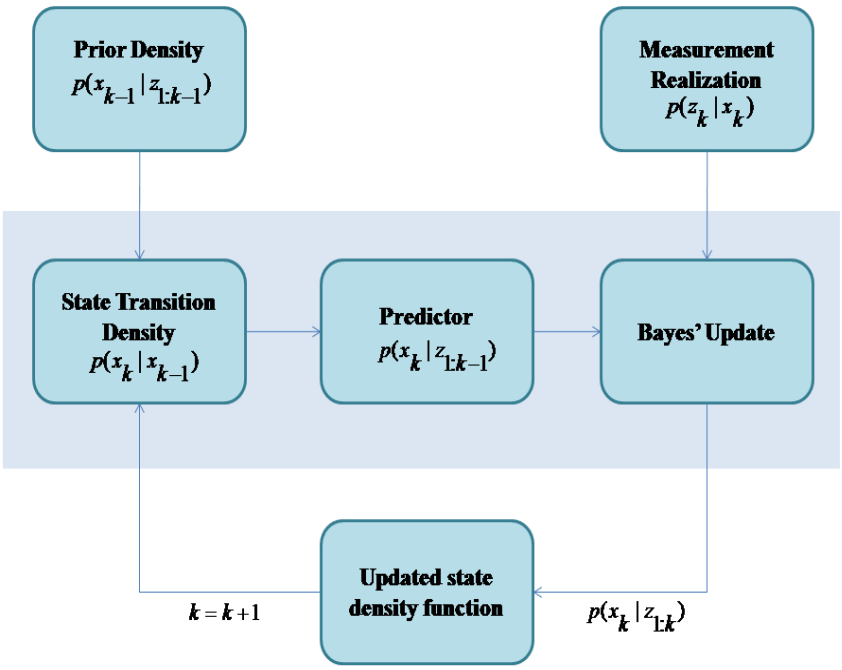

Fig. 1. Predict-update stages in recursive Bayesian estimation

The normalizing constant depends on the likelihood function $p(z_k|x_k)$, defined by the measurement model and the known statistics of measurement noise $\mathbf{n}_k$. These prediction and update equations formulate a *recursive Bayesian* solution for the filtering problem as depicted in Fig. 1.

The recursive relations described above can be easily solved for linear/Gaussian systems. However, if the system under interest is nonlinear/non-Gaussian in nature, the solution requires high dimensional integration and hence making the solution highly intractable. Even storing the values of $p(x)$ for $x \in \mathbb{R}^d$, where d is the number of dimensions, becomes very hard to manage. In such cases, the approximate solution is provided by several non-linear filters. But before discussing about the

non-linear filtering, an understanding of filtering in linear/Gaussian systems is highly recommended. One of the most important filters to be used in the linear/Gaussian case is the *Kalman Filter*, which is the optimal solution when the state space model is linear with uncorrelated Gaussian noise and a Gaussian prior is assumed. This was introduced to the filtering world by Kalman and Bucy in 1961. The rest of this chapter deals with a discussion of the Kalman Filter and its approximated versions for use in non-linear systems.

C. Kalman Filter

The Kalman filter [6] is a linear, discrete-time filter which can be applied to stationary and non-stationary environments recursively. In addition to eliminating the need for storing the entire set of past observed data, the Kalman filter is computationally more efficient than computing the estimate directly. It consists of a set of equations that provides an efficient solution of the least-squares method recursively. It can also provide estimates of the future states, and it can do so when the state of the system is unknown. In other words, it provides the minimum variance estimate of the state of the system by utilizing the information about the dynamic model and the observations that are corrupted by uncorrelated gaussian noise.

Kalman filter assumes that the posterior density at every time step is Gaussian and, hence parameterized by mean and covariance. To be able to apply the Kalman filter, the following assumptions must hold (refer to Equation (2.4) and Equation (2.5) for notation):

- V_k and N_k must be Gaussian random variables with known parameters.
- $f(X_{k-1}, V_k)$ is known and is a linear function of x_{k-1} and v_k.
- $h(X_k, N_k)$ is known and is a linear function of x_k and n_k.

So the state-space model for a system, which holds the above assumptions, can be formulated using the Equation (2.4) and Equation (2.5),

$$x_k = Fx_{k-1} + v_k \tag{2.14}$$

$$z_k = Hx_k + n_k \tag{2.15}$$

where F and H are known matrices defining the linear functions for state and measurement equations respectively. The noise random variables are assumed to be stationary, white Gaussian processes with the zero-mean and the covariances of v_k and n_k are denoted to be Q_k and R_k respectively. Also it is assumed that the sequences $\{V_k\}$ and $\{N_k\}$ are statistically independent,

$$E[v_k v_l^H] = \delta_{kl} Q_k, \quad E[n_k n_l^H] = \delta_{kl} R_k, \quad E[v_k n_l^H] = \mathbf{0}, \ \forall k, l \tag{2.16}$$

The noise random variables that account for both state and measurement models are assumed to be completely uncorrelated.

1. The Kalman Filtering Algorithm

The algorithm follows the two steps, which were already introduced in the recursive Bayesian section, the *prediction* step and the *update* step. In other words, it consists of an iterative prediction-correction process [1]. In the prediction step, the time update is taken where the one-step ahead prediction of observation is calculated; in the correction step, the measurement update is taken where the correction to the estimate of current state is calculated. The steps of the algorithm are summarized next.

The state and measurement models, described by (2.14), will be utilized in evaluating the density function of the state as time updates, $p(x_k|x_{k-1})$, and the measurement likelihood function, $p(y_k|x_k)$, that needs to be updated after a new measurement

is obtained. Since the state and measurement models are linear and are corrupted by Gaussian noise sequences, these required probability density functions will be Gaussian in nature,

$$p(x_k|x_{k-1}) = N(x_k; m_{k|k}, Q_k) \tag{2.17}$$

$$p(y_k|x_k) = N(y_k; \widehat{y_k}, R_k) \tag{2.18}$$

where, $N(x; m, P)$ represents a Gaussian density function with argument x, mean m, and covariance P. Also, the notation used for representing the mean of the state is $m_{i|j} = E[x_i|x_j]$. Mathematically,

$$N(x; m, P) = \frac{1}{\sqrt{2\pi P}} exp\{-\frac{1}{2}(x-m)P^{-1}(x-m)^T\} \tag{2.19}$$

Now, the recursive bayesian solution for this problem can be obtained by using the method described in Eq (2.11) - Eq (2.12),

$$p(x_{k-1}|z_{1:k-1}) = N(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \tag{2.20}$$

$$p(x_k|z_{1:k-1}) = N(x_k; m_{k|k-1}, P_{k|k-1}) \tag{2.21}$$

$$p(x_k|z_{1:k}) = N(x_k; m_{k|k}, P_{k|k}) \tag{2.22}$$

The basic algorithm of the Kalman filter is given below in Table I. Also a pictorial view of the predictor-corrector algorithm for a Kalman filter is given in Fig. 2.

D. Non-linear Filtering

The preceding section of this chapter discusses the filtering techniques when the system model is linear. Now, if a non-linear system model is considered, the techniques used previously may lead to a very poor estimate of the system. There are many methods which are previously used like the Extended Kalman Filtering (EKF), Grid-based

Table I. Kalman filtering algorithm

Initial Variables:

$$m_{0|0} = E[x_0]$$

$$P_{0|0} = E[(x_0 - m_{0|0})(x_0 - m_{0|0})^T]$$

Prior State Densities (at time k):

$$m_{k|k-1} = F_k m_{k-1|k-1}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Measurement Likelihood update:

Innovation term: $S_k = H_k P_{k|k-1} H_k^T + R_k$

Kalman Gain: $K_k = P_{k|k-1} H_k^T S_k^{-1}$

Posterior State Density update:

$$m_{k|k} = m_{k|k-1} + K_k(z_k - H_k m_{k|k-1})$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}$$

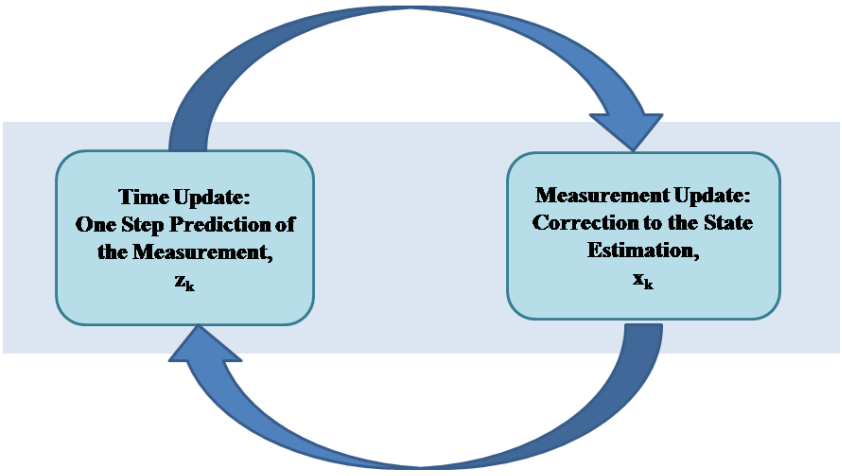

Fig. 2. Predictor-corrector stages in Kalman filtering algorithm

filtering and the Unscented Kalman Filter (UKF). A more recent filtering method, known as the particle filter, is also developed.

The Extended Kalman Filtering is the most commonly used approximate filter. In this method, the system model is linearised and then the standard Kalman Filtering is used on the linearised model. In this kind of filter, the state model and the measurement model, given by Eq (2.4) and Eq (2.5), the equations may be non-linear but differentiable functions. Instead of the non-linear functions for state and measurement equations, a matrix of partial derivatives (the Jacobian) is computed. At each timestep the Jacobian is evaluated with current predicted states thus linearizing the non-linear function around the current estimate. These matrices can be used in the Kalman filter equations like the general linear system model formulated in the preceding section. However, there are some disadvantages with using the extended Kalman filter as it is not an optimal method in estimating the state. Also, if the

initial estimate of the state is poor, then the estimate may not converge due to the error propagation.

Another type of non-linear filter that is a slight improvement to the extended Kalman filter is the Unscented Kalman filter (UKF). In the UKF, the probability densities are approximated by the non-linear transformation of a random variable instead of using the Jacobians. This type of approximation leads to more accurate results than the first-order Taylor expansion of the nonlinear functions in the EKF. The approximation utilizes a set of sample points, which guarantees accuracy with the posterior mean and covariance to the second order for any nonlinearity. It uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points (called sigma points) around the mean. These are then propagated through the non-linear functions and the mean and covariance of the estimate are calculated.

E. Summary

The Kalman filter is the most often used recursive filtering solution in the linear Gaussian case, but for non-linear or non-Gaussian models require additional approximations as explained above. Though a number of approximate filters have been developed for the non-linear/non-Gaussian models, these filters are not optimal under highly non-linear models and all of them suffer from serious drawbacks. These limitations in these filters gave rise to a new class of filters which make use of Monte Carlo methods to high dimensional non-linear models. A number of Monte Carlo filters have been developed across different fields of study. One of such filters is the Particle filter, which will be discussed in the following chapters.

CHAPTER III

THE PARTICLE FILTER

For non-linear dynamic systems, several methods for on-line filtering are introduced in the previous chapter. But the main drawback for these filtering methods is that they are developed based on the local linearization of the nonlinear system equations. For example, the EKF works by linearizing the non-linear system and approximating the noise as Gaussian.

Most dynamical systems in real applications are somewhat non-linear and non-Gaussian in nature. This results in a significant challenge for engineers and scientists to provide an efficient method for real-time estimation and prediction of these systems from sequential observations. Recently, many researchers have begun to consider a new class of filtering methods based on the sequential Monte Carlo (SMC) approach [1,2,3]. The SMC approach can be defined as a set of methods that use a Monte Carlo simulation scheme for solving online estimation and prediction problems.

Sequential Monte Carlo methods have found limited use in the past, except for the last decade, primarily due to their relatively high computational complexity and the lack of adequate computing resources. The fast advances of computers in the recent years and outstanding potential of particle filters have made them a very active area of research recently. The particle filter [3,4,7,8,9,10,11,12] is a sequential Monte Carlo methodology based on the recursive computation of probability distributions. The basic idea of particle filter is to use a number of independent random variables called particles, sampled directly from the state space, to represent the posterior probability, and update the posterior by involving the new observations; the *particle system* is properly located, weighted, and propagated recursively according to the Bayesian rule.

The particle filters can be applied to any state space model and it generalizes the Kalman filter. The advantage of particle filtering over other methods is in that the chosen approximation does not involve linearizations around current estimates but rather approximations in the representation of the desired distributions by discrete random measures. The particle filters are best suited for non-linear state-space models with non-Gaussian noise. They have found application in many areas such as channel equalization, estimation and coding, wireless channel tracking, artificial intelligence, speech enhancement, speech recognition, and machine learning.

A. Sequential Importance Sampling

In order to make Bayesian importance sampling more practical, it will be convenient to calculate the particle weights recursively. The *sequential importance sampling (SIS)* [2,3,4,7,8,11] algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past decades. It is a technique for implementing a recursive Bayesian filter by MC simulations. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

Let $\{x^i_{0:k}, w^i_k\}$ define a random discrete measure that approximates the posterior pdf $p(x_{0:k}|z_{1:k})$. Where, $\{x^i_{0:k}; i = 1, ..., N_s\}$ is a set of sample points with associated weights $\{w^i_k; i = 1, ..., N_s\}$ and $x_{0:k}$ is the set of all states up to time k. The weights are normalized such that $\sum_i w^i_k = 1$. By SIS algorithm, the set $\{x^i_{0:k}, w^i_k\}$ is recursively computed from the set $\{x^i_{0:k-1}, w^i_{k-1}\}$, when a new measurement z_k is available at time

k. Specifically, suppose at time $k-1$ the posterior pdf $p(x_{0:k-1}|z_{1:k-1})$ is approximated by a random measure $\{x^i_{0:k-1}, w^i_{k-1}\}$, then SIS algorithm builds a random measure by appending newly generated particles x^i_k to the $x^i_{0:k-1}$ and updating the weights w^i_k to form $\{x^i_{0:k}, w^i_k\}$ that properly represent the posterior pdf $p(x_{0:k}|z_{1:k})$. Then, the posterior density at time k is approximated as [2]

$$p(x_{0:k}|z_{1:k}) = \sum_{i=1}^{N_s} w^i_k \delta(x_{0:k} - x^i_{0:k}). \tag{3.1}$$

The above equation represents the discrete weighted approximation to the true posterior, $p(x_{0:k}|z_{1:k})$. The weights can be chosen using the principle of importance sampling. If the samples $x^i_{0:k}$ were drawn from an importance density $q(x_{0:k}|z_{1:k})$, the weights are given by

$$w^i_k \propto \frac{p(x^i_{0:k}|z_{1:k})}{q(x^i_{0:k}|z_{1:k})}. \tag{3.2}$$

At each iteration by using the approximated $p(x_{0:k-1}|z_{1:k-1})$, and with a new set of samples; the pdf $p(x_{0:k}|z_{1:k})$ is calculated. The importance density $q(x_{0:k}|z_{1:k})$ is factorized as

$$\begin{aligned} q(x_{0:k}|z_{1:k}) &= \frac{q(x_{0:k}, z_{1:k})}{q(z_{1:k})} \\ &= \frac{q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}, z_{1:k})}{q(z_{1:k})} \\ &= \frac{q(x_k|x_{0:k-1}, z_{1:k})q(z_k|x_{0:k-1}, z_{1:k-1})q(x_{0:k-1}, z_{1:k-1})}{q(z_k|z_{1:k-1})q(z_{1:k-1})} \\ &= q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}) \end{aligned} \tag{3.3}$$

By the equation (3.3), the samples $x^i_{0:k} \sim q(x_{0:k}|z_{1:k})$ are obtained by augmenting each of the existing samples $x^i_{0:k-1} \sim q(x_{0:k-1}|z_{1:k-1})$ with the new state

$x_k^i \sim q(x_k|x_{0:k-1}, z_{1:k})$. Then the pdf $p(x_{0:k}|z_{1:k})$ is expressed as,

$$\begin{aligned}
p(x_{0:k}|z_{1:k}) &= \frac{p(x_{0:k}, z_{1:k})}{p(z_{1:k})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}, z_{1:k-1})}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_k|x_{0:k-1}, z_{1:k-1})p(x_{0:k-1}, z_{1:k-1})}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_k|x_{0:k-1}, z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
&= \frac{p(z_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
p(x_{0:k}|z_{1:k}) &\propto p(z_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|z_{1:k-1}) \qquad (3.4)
\end{aligned}$$

where $p(z_k|z_{1:k-1})$ is a normalized constant. Now substituting the equations (3.3) and (3.4) in equation (3.2), then we have

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})}. \qquad (3.5)$$

Furthermore, if $q(x_k^i|x_{0:k-1}^i, z_{1:k}) = q(x_k^i|x_{k-1}^i, z_k)$, then the *importance density* depends only on x_{k-1} and z_k. This is particularly useful in the common case when only a filtered estimate of $p(x_k|z_{1:k})$ is required for each time step. Then the modified weight is given by

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \qquad (3.6)$$

Then the required posterior filtered density $p(x_k|z_{1:k})$ is given by

$$p(x_k|z_{1:k}) = \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i). \qquad (3.7)$$

Thus the SIS algorithm consists of recursive propagation of weights and samples as each measurement is received sequentially. A pseudo-code description of the SIS algorithm is given below [2, 3].

Algorithm 1 SIS Particle Filtering Algorithm

1: **procedure** SIS(x^i_{k-1}, w^i_{k-1})
2: **for** i := 1 to N_s **do**
3: Draw $x^i_k \sim q(x_k|x^i_{k-1}, z_k)$
4: Assign each particle with the importance weights according to Eq (3.6)
5: **end for**
6: Normalize all the importance weights so that they add up to unity.
7: **end procedure**

B. Degeneracy Phenomenon and Resampling in Particle Filters

In particle filters, the posterior probability is represented by a set of randomly chosen weighted samples drawn from an importance density. However a common problem with the sequential importance sampling is that after a few iterations, most particles will have negligible weight. It means that the weight is concentrated on certain particles only. This problem is called *degeneracy problem* [2,3,11,12].

The variance of the importance weights increases over time, thus making it impossible to avoid the degeneracy problem [7]. Effectively a large computational effect is devoted to updating particles whose contribution to approximate the posterior pdf is almost zero. A suitable measure of degeneracy of the algorithm is the *effective sample size* [7,13], which is given by

$$N_{eff} = \frac{N_s}{1 + Var(w^{*i}_k)}. \tag{3.8}$$

where, $w^{*i}_k = p(x^i_k|z_{1:k})/q(x^i_k|x^i_{k-1}, z_k)$. Thus the effective sample size cannot be evaluated exactly, an estimate is calculated instead which is given by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}(w^i_k)^2}. \tag{3.9}$$

A small N_{eff} implies a severe degeneracy. There are three basic measures to mitigate the degeneracy problem in particle filters,

1. by increasing the number of samples Ns,

2. resampling,

3. by a good choice of importance density.

The simplest method to mitigate the degeneracy effect is to use a very large N_s. However it will result in a drastic increase in the computational load on the system. The next section in this chapter discusses the method of resampling and various techniques based on this method.

1. Resampling

Effects of degeneracy in particle filter is reduced by using resampling [13,17,18,19,20], where the particles having small weights are eliminated and the particles with large weights are replicated. At every step, the effective particle size is calculated. The calculated effective size is compared with the predefined threshold, based on that the resampling step will be carried out. All the particles after resampling have the same weight $1/N$. By this, the particles having large weight are repeated and particles having less weight are eliminated. Thus, the samples are concentrated in the region of interest. The resampling stage is depicted in the Fig. 3. From Fig. 3, it may be seen that the diameters of the circles are proportional to the weights of the particles and after resampling all the particles are having the same weight.

The resampling method can be briefly explained as follows. At every step, by comparing the covariance of a set of samples drawn from the posterior and the covariance obtained through the use of importance sampling a measurement of the

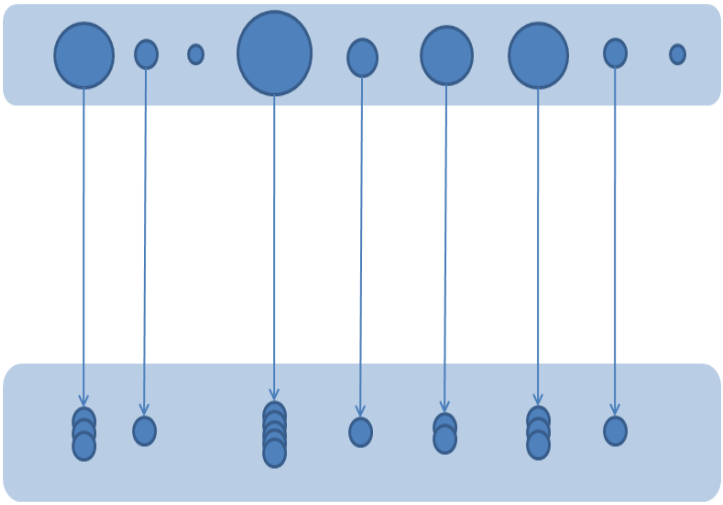

Fig. 3. Particle resampling with weights represented by size

sampling efficiency is obtained which in turn will give an expression for the effective sample size. The calculated effective size is compared with the predefined threshold and the resampling step is carried out based on that. The resampling stage involves drawing of N samples from the *a posteriori* pdf with replacement. All the particles after resampling will be assigned equal weight, $1/N$. With this type of assignment, the particles that have a large weight will be repeated a higher number of times and particles having less weight are eliminated. Thus the samples will become more concentrated in the region of interest. Fig. 3 explains this method pictorially.

Resampling involves a mapping of random measure $\{x_k^i, w_k^i\}$ to $\{x_k^{*i}, 1/N\}$. The set of random samples $\{x_k^{*i}\}$ is generated by resampling (with replacement) N times from an approximate discrete representation of $p(x_k|z_{1:k})$ with a probability, $p(x_k^{*i} = x_k^j) = w_k^j$. The resulting sample is an i.i.d sample from the posterior density $p(x_k|z_{1:k})$. In Fig. 4, the acronym CSW stands for the cumulative sum of weights of the random measure $\{x_k^i, w_k^i\}$, and random variable u_i is uniformly distributed in the interval $[0, 1]$. From Fig. 4, the main idea in the process of resampling is to select the new

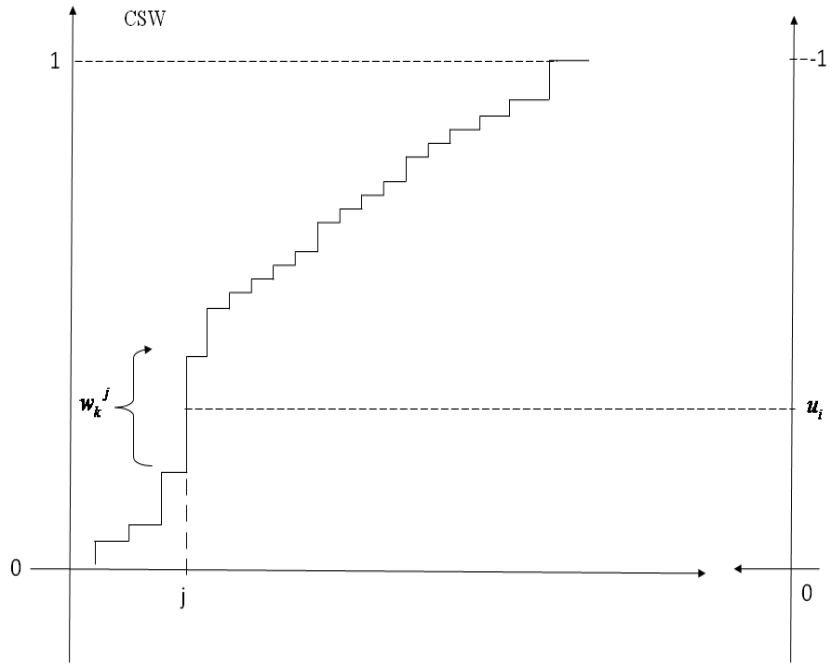

Fig. 4. Resampling process for a particle filter

particles by comparing an ordered set of uniformly distributed random numbers u_i lies in the interval $[0, 1]$ with the cumulative sum of the normalized weights. It may be seen that from Fig. 4 that the uniform random variable u_i maps into index j and the corresponding particle x_k^j has a good chance of being selected and multiplied because of its high value of w_k^j. Two of the most efficient resampling techniques, systematic resampling and residual resampling, are discussed in the next section.

2. Systematic Resampling

In systematic resampling [18,19], the interval is divided in N strata and one sample is taken from every stratum as in stratified sampling, but the samples are no longer independent: all the samples have the same position within a stratum. This gives the minimal discrepancy for N samples. The algorithm is summarized below:

Algorithm 2 Systematic Resampling Algorithm

1: Draw $x_k^i \sim q(x_k|x_{k-1}^i, z_k)$

2: Obtain the N ordered random numbers u_k using, $u_k = \frac{(k-1)+\tilde{u}}{N}$

3: Allocate the n_i copies of the particle x_i to the new distribution, n_i = the number of $u_k \in [\sum_{s=1}^{i-1} w_s, \sum_{s=1}^{i} w_s)$

3. Residual Resampling

Residual resampling [18,19] uses a somewhat different approach to resample. The idea is that a large part of the number of offspring n_i can be determined without resorting to random numbers. This can be achieved by taking the integer part of Nw_i. To retain the original population size some more copies need to be made. These residual particles are randomly selected with replacement from the original particles using modified weights. The algorithm is summarized below:

Algorithm 3 Residual Resampling Algorithm

1: Allocate $n_i' = \lfloor Nw_i \rfloor$ copies of particle x_i to the new distribution

2: Resample $m = N - \sum n_i'$ particles from $\{x_i\}$ by making n_i'' copies of x_i where the probability for x_i is proportional to $w_i' = Nw_i - n_i'$

4. Generic Particle Filter

An algorithm of a generic particle filter based on the SIS method and resampling technique discussed before is given below [2,3].

In this algorithm, the choice of the proposal or importance distribution is the most critical design issue. It starts by sampling samples from the prior distribution and calculates the first set of weights from the first measurement that is available.

Algorithm 4 Generic Particle Filtering Algorithm

1: **procedure** PF(x^i_{k-1}, w^i_{k-1})
2: **for** i := 1 to N_s **do**
3: Draw $x^i_k \sim q(x_k|x^i_{k-1}, z_k)$
4: Assign each particle with the importance weights according to Eq (3.6)
5: **end for**
6: Normalize all the importance weights so that they add up to unity.
7: Calculate effective sample size N_{eff} by using the Eq (3.9)
8: **if** $N_{eff} < N_T$ **then**
9: Resample using systematic or residual technique
10: **end if**
11: **end procedure**

This step accounts for the predictor stage of the algorithm. In each iteration the samples are drawn according to a selected importance distribution. Then, the weights are updated by using the selected proposal distribution and the drawn samples. This steps represents the update stage. A pictorial representation of this algorithm can be depicted as shown in Fig. 5.

From Fig. 5, it can be seen that the particles are modified by the importance density function. The higher the probability, the denser the particles become concentrated. The effective size of all the particles is calculated. If the effective size is less than the predefined threshold, then the resampling step is carried out. After resampling, all the particles will be assigned the same weight. Then the Particle Filter algorithm is applied using the new particles to progress through to the next stage of particle system.

Although the resampling step reduces the effects of the degeneracy problem, it

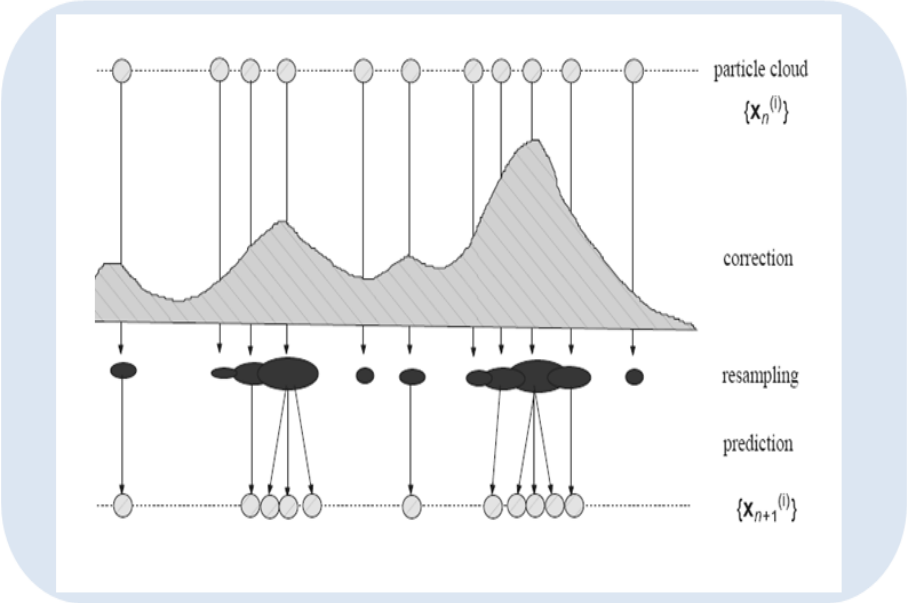

Fig. 5. Generic particle filter with resampling

introduces other problems. First, it limits the opportunity to parallelize the implementation since all the particles must be combined. Second, the particles that have high weights are statistically selected many times, this lead to a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem is known as *sample impoverishment* [1,3]. There are techniques namely *Markov chain Monte Carlo* (MCMC) [3], *regularization* [3] method to reduce the effect of sample impoverishment.

C. Choice of Importance Density

The choice of the sampling density of the algorithm affects the quality of the state estimate significantly [2,3,12]. However there are number of choices for the sampling density. The sampling density must fulfill a criterion to ensure convergence of the estimates as number of samples N_s becomes large. Further, the shape of the sampling density must be as close to the true filtering pdf as possible and it should guarantee

a minimum variance. The sampling density should also be as simple with respect to the weights evaluation as possible. The most often used sampling density is the Prior Density, $p(x_k|x_{k-1})$.

1. Prior Importance Function

This sampling density is frequently used due to its simplicity and easy weight computation. Here the current estimate is ignored during drawing of samples and thus low quality estimates will be obtained. The prior sampling density takes the form [2,3],

$$q(x_k^i|x_{k-1}^i, z_k) = p(x_k^i|x_{k-1}^i) \tag{3.10}$$

Substituting this equation in the Eq (3.6), we get,

$$w_k^i \propto w_{k-1}^i p(z_k|x_k^i) \tag{3.11}$$

If the transitional prior, $p(x_k^i|x_{k-1}^i)$, is used as the importance density and if it has a much broader distribution than the likelihood function, $p(z_k|x_k)$, then only a few particles will be assigned a high weight. Consequently, the particles will degenerate rapidly and the filter does not work. The particles should be in the right place (in the regions of high likelihood) by incorporating the current observation, then only efficient estimate is obtained through the particle filter algorithm.

2. Optimal Sampling Density

Instead of sampling the state from the prior distribution, a more evenly distributed set of weights can be sampled by using a different proposal density or importance function. If such a sampling density is chosen to minimize the variance of weights [7], so that effective sample size is maximized, then it is said to be optimal sampling

density. This sampling density will then assume the form,

$$\begin{aligned} q(x_k^i|x_{k-1}^i, z_k)_{opt} &= p(x_k^i|x_{k-1}^i, z_k) \\ &= \frac{p(x_k^i, x_{k-1}^i, z_k)}{p(x_{k-1}^i, z_k)} \\ &= \frac{p(z_k|x_k^i, x_{k-1}^i)p(x_k^i, x_{k-1}^i)}{p(z_k|x_{k-1}^i)p(x_{k-1}^i)} \\ q(x_k^i|x_{k-1}^i, z_k)_{opt} &= \frac{p(z_k|x_k^i, x_{k-1}^i)p(x_k^i|x_{k-1}^i)}{p(z_k|x_{k-1}^i)} \end{aligned} \tag{3.12}$$

Substituting this equation in the Eq (3.6), we get,

$$w_k^i \propto w_{k-1}^i \int p(z_k|x_k')p(x_k'|x_{k-1}')dx_k' \tag{3.13}$$

The above chosen optimal density has two limitations. It requires sampling from the pdf $p(x_k^i|x_{k-1}^i, z_k)$ and the evaluation of integral expression (3.13) which cannot be done easily. When x_k belongs to a finite set, then the integral expression become a sum, and sampling from the optimal importance density becomes possible.

Consider the case where the state dynamics is nonlinear, the measurement equation is linear, and all the random elements in the model are additive Gaussian. It can be shown that in this case, both the optimal importance density and the likelihood equation are Gaussian. The proof for such a model is shown in Appendix A.

D. Sampling Importance Resampling (SIR) Particle Filter

The Sampling Importance Sampling (SIR) appproach proposed by Gordon [2,3,4] is illustrated in this section. The SIR filter is a special case of the SIS algorithm. It is a Monte Carlo method that can be applied to recursive Bayesian filtering problems. The SIR algorithm is rather straightforward and can be derived easily from the SIS algorithm by an appropriate choice of the importance density and the resampling

step. Here we use the prior transition density, $p(x_k^i|x_{k-1}^i)$, as the importance function owing to its convenient usage and easy computation. The resampling is done at every step of the time index instead of comparing it to a threshold as is the case with other particle filters. The algorithm for a simple SIR particle filter is given below.

Algorithm 5 SIR Particle Filtering Algorithm

1: **procedure** SIRPF(x_{k-1}^i, w_{k-1}^i)
2: **for** i := 1 to N_s **do**
3: Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
4: Assign each particle with the importance weights according to Eq (3.6)
5: **end for**
6: Normalize all the importance weights so that they add up to unity.
7: Resample using systematic or residual technique
8: **end procedure**

E. Methods of Improving Particle Filters

Many particle filter algorithms have been proposed by various scientists and engineers to compensate for the drawbacks of the particle degeneracy and sample impoverishment which are the major bottlenecks for the particle filters. Some of the methods that have been used to improve the performance of the particle filter include the following.

1. Choice of Proposal Distribution

A first method for choosing an optimal importance density involves in maximizing the effective sample size N_{eff}. In this method, the optimal density function is chosen

such that it minimizes the variance of the weights as the time index progresses. However, the calculation of the optimal important density requires to evaluate an multi-dimensional integral that is discussed in the previous section.

2. Local Linearization

An optimal importance density can be approximated by using the most current measurement through a set of the standard nonlinear filters. The approximated density propagates the particles towards the likelihood function and consequently the hybrid particle filter performs better than the SIR filter.

3. Regularization

Though the resampling reduces the effects of the degeneracy phenomena, it causes other practical problems, as discussed before, called sample impoverishment. A modified particle filtering algorithm in which the resampling process is performed upon a kernel-based density estimation can be a potential solution to handle the sample impoverish effect.

4. MCMC Move

Markov chain Monte Carlo (MCMC) methods provide a relatively easy way of generating samples from any probability distribution. It can also be a potential solution to the sample impoverishment in resampling step as well as the regularization scheme.

5. Rao-Blackwellization

In some cases, the components of the model may have linear dynamics and can be well estimated using a conventional Kalman filter. The Kalman filter is combined with a particle filter to reduce the number of particles needed to obtain a given level

of performance. This method can reduce the variance of the MC estimates as well as the number of samples.

F. Simulation Results

The following nonlinear state space model is considered for the simulation of sampling importance resampling (SIR) filter, which is given by [2]

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + v_{k-1} \tag{3.14}$$

$$z_k = \frac{x_k^2}{20} + n_k \tag{3.15}$$

From the state space model (3.14), the prior density $p(x_k|x_{k-1})$ and the likelihood function $p(z_k|x_k)$ are respectively given by

$$p(x_k|x_{k-1}) = N(x_k; f_k(x_{k-1}, k), Q_{k-1}) \tag{3.16}$$

$$p(z_k|x_k) = N(z_k; \frac{x_k^2}{20}, R_k) \tag{3.17}$$

It is assumed that the noise random variables n_k and v_{k-1} are zero mean Gaussian random variables with variances Q_{k-1} and R_k respectively. For the simulation of SIR filter in the MATLAB environment, the following parameters are used.

- Noise variances are and respectively.
- Number of states M =100
- Number of particles N=50,100
- Number of Monte Carlo runs=1000

The samples $\{x_k^i\}_{i=1}^N$ and the corresponding weights $\{w_k^i\}_{i=1}^N$ are generated using the SIR particle filter algorithm discussed in the previous section. The estimate of the

state at each time instant k is calculated by using the set of samples and corresponding weights, which is given by the sum of products of samples and corresponding weights.

$$x_{k_{est}} = \sum_{i=1}^{N} x_k^i w_k^i \tag{3.18}$$

To obtain a performance measure on the process of state estimation, the Root Mean Square Error (RMSE) measure of the true state with respect to the estimated state is computed. It is given by the expression

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_k - x_{k_{est}})}. \tag{3.19}$$

Fig. 6 shows 100 true values of the state x_k as a function of time k. Fig. 7 shows the

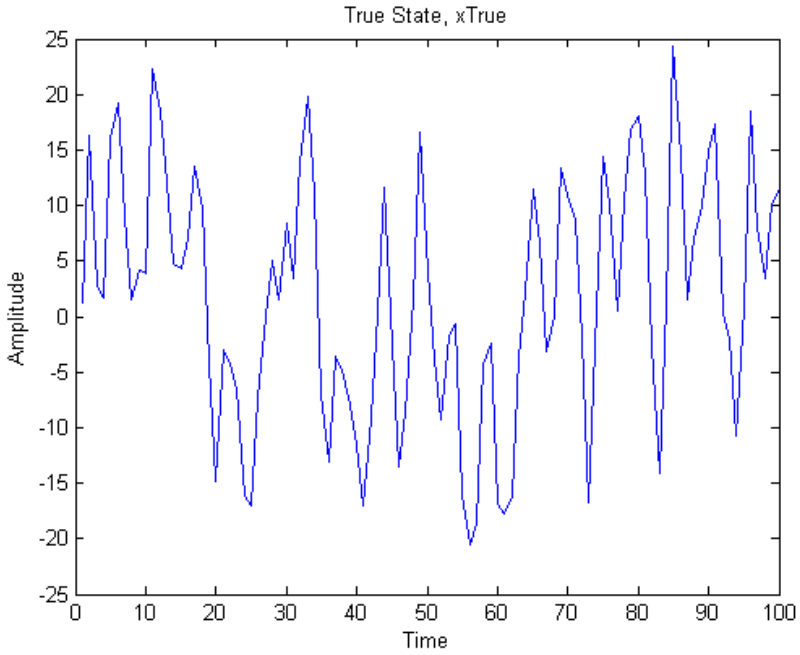

Fig. 6. True state x_k as a function of time k

100 measurements z_k as a function of time k.

Fig. 8 shows the estimated state and true state for comparison. In this case, SIR

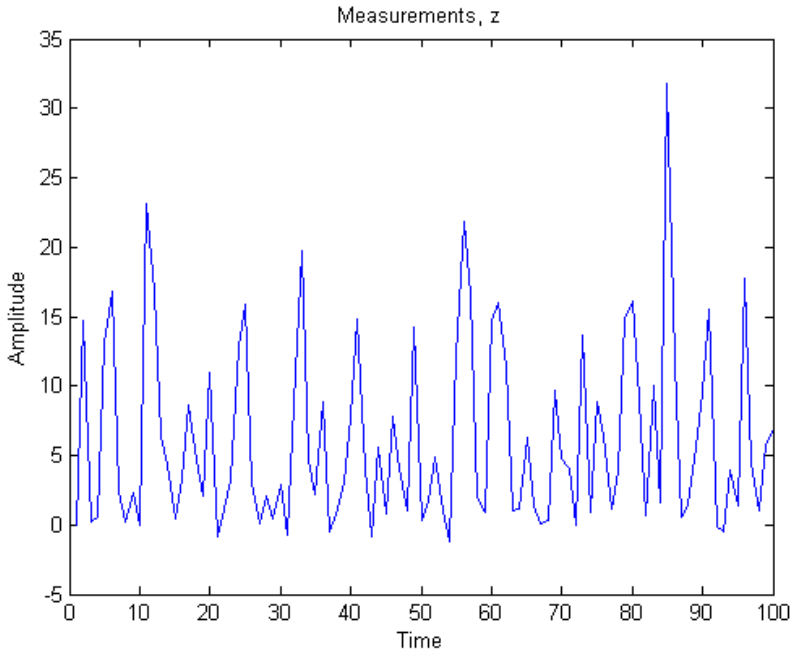

Fig. 7. Measurements z_k as a function of time k

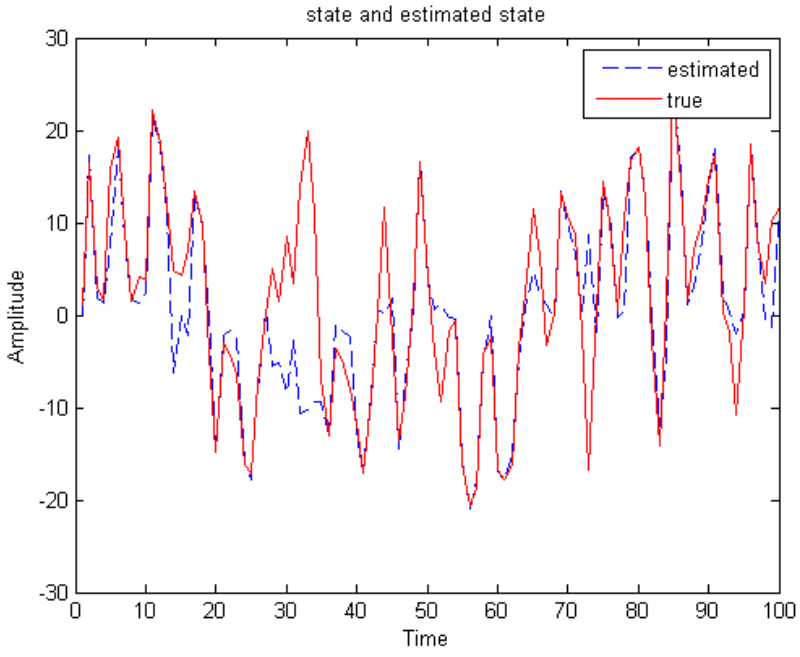

Fig. 8. True state x_k and estimated state x'_k using 50 particles

filter uses 50 particles for estimating the state. The RMSE of SIR filter is obtained by averaging over 1000 independent realizations and is found to be 9.6144.

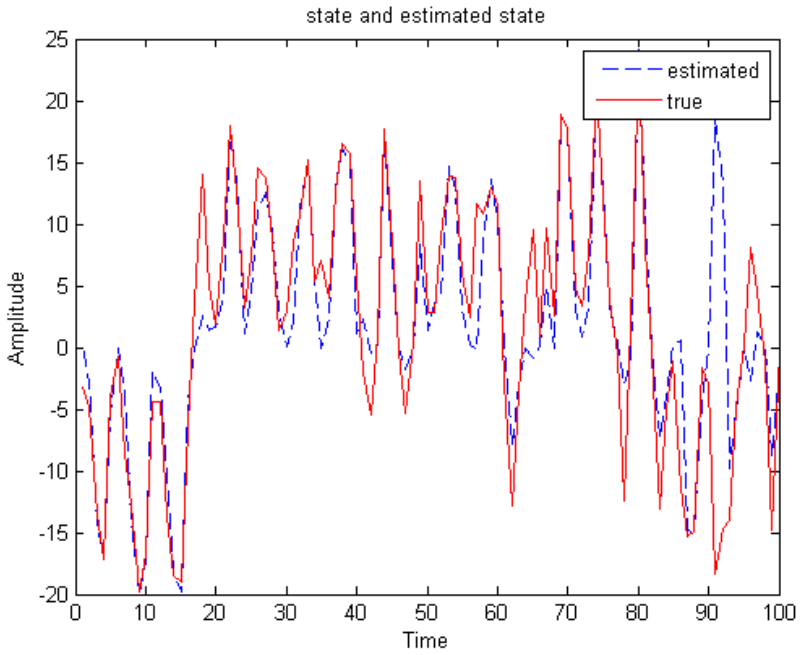

Fig. 9. True state x_k and estimated state x'_k using 100 particles

Fig. 9 shows the estimated state of the SIR filter when 100 particles are used. For comparison, we have also plotted the true states. It may be noted here that there is a close similarity between the true states and estimated states by SIR filter. The RMSE of SIR filter is found to be 5.9006, which represents for a considerable improvement over the RMSE for 50 paricles only.

It is observed that there is a very high degree of improvement in the RMSE when 100 particles are used. So, the higher the number of particles used the lower will be the RMSE.

CHAPTER IV

DETECTION IN FLAT-FADING CHANNEL USING PARTICLE FILTER

In the transmission of digital information over a communication channel, which is fading dispersive, the interference to the signal is mainly caused by delayed versions of the original signal [21,22]. The optimal detection scheme for such a channel, with known characteristics, is the maximum-likelihood sequence estimation (MLSE) detector [21,23]. It finds the best symbol vector that minimizes the Euclidean distance with respect to the received signal, but its complexity increases exponentially with the dimension of the parameter to be estimated. There are other alternatives, which require only linear complexity, like the zero-forcing (ZF) detector and the minimum mean square error (MMSE) detector [21]. However, they provide only sub-optimal performance with respect to the error probabilities. Most of sub-optimal algorithms include a two stage receiver structure with a channel estimation stage followed by a sequence detection stage. J.K. Cavers [24] suggested a pilot method for detection of signals in fading channels. But the transmission of pilot requires bandwidth and will decrease the communication throughput causing significant overhead problem. A novel adaptive Bayesian receiver for signal detection and decoding in fading channels with unknown channel statistics is presented in [25]. It is based on the sequential Monte Carlo methodology that has recently emerged in the field of statistics. The basic idea is to treat the transmitted signals as missing data and to sequentially impute multiple samples of them based on the observed signals. The imputed signal sequences, together with their importance weights, provide a way to approximate the Bayesian estimate of the transmitted signals [25]. This SMC technique easily handles the non-Gaussian ambient channel noise, without the use of any training/pilot symbols or decision feedback. In this chapter, a derivation of the state space model

of SISO system when fading coefficients are modeled by both auto regressive-moving average (ARMA) and auto regressive (AR) processes is presented. The derivation of an efficient particle filter algorithm for such a problem is presented later. Also the application of the residual resampling algorithm and effect of the delayed estimation approach are also discussed with the simulation results at the end.

A. Signal Model

Consider a communication system signalling through a flat fading channel with additive ambient noise as given in Fig. 10 [17,25]. As Fig. 10 shows, the input binary

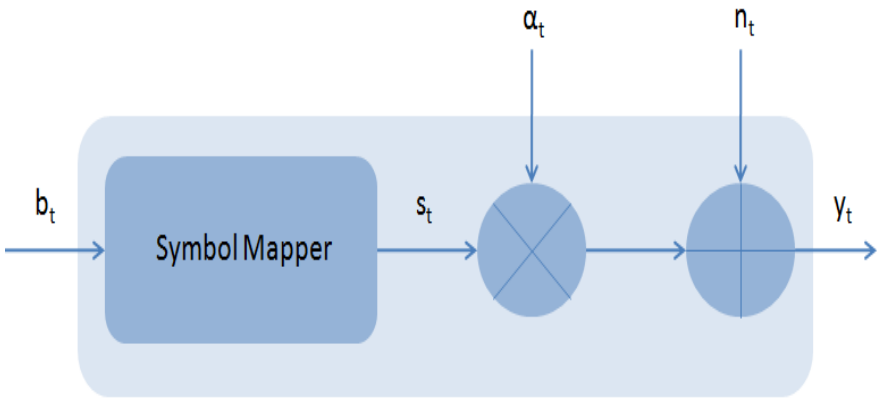

Fig. 10. Communication system model over a flat fading channel

information bits $\{b_t\}$ are passed to a symbol mapper yielding complex data symbols $\{s_t\}$, which take a finite value from the alphabet set $A = \{a_1, a_2, ..., a_{|A|}\}$. Each symbol is transmitted through a flat-fading channel, where it is multiplied by a fading channel coefficient with the addition of ambient channel noise. The received signal y_t is given by

$$y_t = \alpha_t s_t + n_t \tag{4.1}$$

as shown in [17,25], where, α_t represents the fading channel coefficient at time t, s_t represents the transmitted symbol at that time, and n_t represents the additive noise of the channel at that time. Here $t = 0, 1, ...$ These processes $\{\alpha_t\}, \{s_t\}, and \{n_t\}$ are assumed to be mutually independent. It is assumed that the additive noise n_t in Eq (4.1) is a sequence of independent and identically distributed (i.i.d) zero-mean complex random variables.

In this work, two types of noise distributions are considered. In the first type, $\{n_t\}$ assumes a complex Gaussian distribution of zero mean and variance σ^2 given by

$$n_t \sim N_c(0, \sigma^2). \tag{4.2}$$

In the second type of distribution, $\{n_t\}$ assumes Middeleton Class A noise model [9,26] for modeling a non-Gaussian distribution. This particular noise model has been extensively used for physical noise in radio and acoustic channels. Here, $\{n_t\}$ takes the form of a two-term mixture Gaussian distribution [9]

$$n_t \sim (1 - \epsilon)N_c(0, \varsigma^2) + \epsilon N_c(0, k\varsigma^2), \tag{4.3}$$

where, $N_c(0, \varsigma^2)$ represents the normal ambient noise, $N_c(0, k\varsigma^2)$ represents an impulsive component, ϵ is the probability that impulsive pulses can occur, and k is positive integer with $k > 1$.

It is further assumed that the channel-fading process is Rayleigh i.e., the fading coefficients $\{\alpha_t\}$ form a complex Gaussian process [25]. Also the fading process is modelled here by the output of a Butterworth filter driven by white Gaussian noise.

1. Fading Coefficients as ARMA Process

The generalized form of an ARMA process of order (r, r) is given by

$$\phi_r \alpha_{t-r} + ... + \phi_1 \alpha_{t-1} + \alpha_t = \theta_0 u_t + \theta_1 u_{t-1} + ... + \theta_r u_{t-r} \tag{4.4}$$

where $\{u_t\}$ is a white complex Gaussian noise sequence with independent real and complex components. The ARMA coefficients, $\{\phi_i\}$, $\{\theta_i\}$, and the order r of the Butterworth filter, are chosen so that the transfer function of the filter matches the power spectral density of the fading process, which in turn, is determined by the channel Doppler frequency [25]. By assuming that the statistical properties of the fading process are known *a priori*, the order and the coefficients of the Butterworth filter are known. Define the state variable x_t such that

$$x_t = -\phi_r \alpha_{t-r} - .... - \phi_1 \alpha_{t-1} - \alpha_t + u_t. \tag{4.5}$$

By writing the Eq (4.5) in matrix form, we get,

$$\begin{bmatrix} x_t \\ x_{t-1} \\ : \\ : \\ x_{t-r} \end{bmatrix} = \begin{bmatrix} -\phi_1 & -\phi_2 & ... & -\phi_r & 0 \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ : & : & ::: & : & 0 \\ 0 & 0 & ::: & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ : \\ : \\ x_{t-r-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ : \\ : \\ 0 \end{bmatrix} u_t$$

Let $\mathbf{x}_t \triangleq [x_t \ x_{t-1} \ x_{t-2} \ ... \ x_{t-r}]^T$. From the matrix form the state equation can be shown as

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.6}$$

where, $u_t \sim N_c(0, 1)$. Now, from Eq (4.5) and Eq (4.4), we get

$$\alpha_t = \theta_0 x_t + \theta_1 x_{t-1} + ... + \theta_r x_{t-r}. \tag{4.7}$$

The Eq (4.7) can be written in matrix form as shown below

$$\alpha_t = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & ... & \theta_r \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ : \\ : \\ x_{t-r} \end{bmatrix}$$

Define $\mathbf{h} = [\theta_0 \ \theta_1 \ \theta_2 \ ... \ \theta_r]^H$. Then the value of α_t is given by

$$\alpha_t = \mathbf{h}^H \mathbf{x}_t. \tag{4.8}$$

Using Eq (4.8) and Eq (4.1), the state-space model for the system with additive Gaussian noise can be formulated as:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.9}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t \tag{4.10}$$

where $\{v_t\}$ is a white complex Gaussian noise sequence with unit variance and independent real and imaginary components. If the additive noise in Eq (4.1) is non-Gaussian and is modeled by Eq (4.3), then an indicator random variable I_t is used to model the state-space model. The indicator variable is defined by

$$I_t = \begin{cases} 1, & \text{if } n_t \sim N_c(0, \varsigma^2) \\ 2, & \text{if } n_t \sim N_c(0, k\varsigma^2) \end{cases} \tag{4.11}$$

with $p(I_t = 1) = (1 - \epsilon)$ and $p(I_t = 2) = \epsilon$. Let $\sigma_1^2 = \varsigma^2$ and $\sigma_2^2 = k\varsigma^2$, then the state-space model can be given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.12}$$

$$y_t = s_t\mathbf{h}^H\mathbf{x}_t + \sigma_{I_t}v_t \tag{4.13}$$

2. Fading Coefficients as an AR Process

The generalized form of an Autoregression process of order r is given by

$$\phi_r\alpha_{t-r} + ... + \phi_1\alpha_{t-1} + \alpha_t = u_t \tag{4.14}$$

where $\{u_t\}$ is a white complex Gaussian noise sequence with independent real and complex components; $\{\phi_i\}$ are called the AR coefficients [25].

Let $\mathbf{x}_t \triangleq [x_t \;\; x_{t-1} \;\; x_{t-2} \;\; ... \;\; x_{t-r}]^T$. Using a similar procedure as in the ARMA process, we have the following state-space form:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.15}$$

where,

$$F = \begin{bmatrix} -\phi_1 & -\phi_2 & ... & -\phi_r & 0 \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ : & : & ::: & : & 0 \\ 0 & 0 & ::: & 1 & 0 \end{bmatrix}$$

$$g = \begin{bmatrix} 1 & 0 & ... & 0 & 0 \end{bmatrix}^T$$

and $u_t \sim N_c(0, 1)$. Now, from Eq (4.5) and Eq (4.14), we get,

$$\alpha_t = x_t \tag{4.16}$$

The Eq (4.16) can be written in matrix form as shown below,

$$\alpha_t = \begin{bmatrix} 1 & 0 & 0 & ... & 0 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ : \\ : \\ x_{t-r} \end{bmatrix}$$

Define $\mathbf{h} = [1\ 0\ 0\ ...\ 0]^H$. Then the value of α_t is given by

$$\alpha_t = \mathbf{h}^H \mathbf{x}_t \tag{4.17}$$

Using Eq (4.17) and Eq (4.1), the state-space model for the system with additive Gaussian noise can be formulated as:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.18}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t \tag{4.19}$$

where $\{v_t\}$ is a white complex Gaussian noise sequence with unit variance and independent real and imaginary components.

Similarly for the non-Gaussian case, we have,

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{4.20}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma_{I_t} v_t \tag{4.21}$$

B. Particle Filtering Method

Consider the flat-fading channel with additive Gaussian noise given by Eq (4.9). Let $\mathbf{Y}_t = (y_1, y_2, ..., y_t)$ be the received data and $\mathbf{S}_t = (s_1, s_2, ..., s_t)$ be the transmitted data up to time t respectively. **Statement of the Problem:** To estimate the a

posteriori probabilities of the information symbols

$$p(s_t = a_i|\mathbf{Y}_t), \quad a_i \in A$$

based on the received signals $\mathbf{Y}_t$ and the *a priori* symbol probabilities $p(s_t = a_i)$, without the knowledge of channel coefficients $\alpha_t = \mathbf{h}^H\mathbf{x}_t$. Consider M-ary phase-shift keying (MPSK) signals are transmitted i.e.,

$$a_i = \exp(j\frac{2\pi i}{|A|}), \qquad i = 0, 1, ..., |A| - 1 \tag{4.22}$$

Assume that the transmitted symbols are independent and equiprobable i.e.,

$$\begin{aligned} p(s_t = a_i|\mathbf{S}_{t-1}) &= p(s_t = a_i), \qquad a_i \in A \\ p(s_t = a_i) &= \frac{1}{|A|}, \qquad i = 0, 1, ..., |A| - 1 \end{aligned} \tag{4.23}$$

In order to implement the particle filter (with m particles), a set of Monte Carlo samples of the transmitted symbols $\{\mathbf{S}_t^{(j)}\}_{j=1}^m$ with its corresponding importance weights $\{w_t^{(j)}\}_{j=1}^m$, properly weighted with respect to the distribution $p(\mathbf{S}_t|\mathbf{Y}_t)$, are to be computed. Let the optimal sampling density, discussed in the previous chapter, as the importance function. We have

$$q(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_t) = p(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_t). \tag{4.24}$$

With this choice of sampling density, the importance weights are updated according to the equation

$$w_t^{(j)} \propto w_{t-1}^{(j)} \sum_{a_i \in A} \rho_{t,i}^{(j)} \tag{4.25}$$

where, $\rho_{t,i}^{(j)} = p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(s_t = a_i)$. This is obtained by a similar derivation discussed in the section on Importance sampling densities in the previous chapter.

Now, consider the expression for sampling density,

$$\begin{aligned}
p(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_t) &= p(s_t|\mathbf{S}_{t-1}^{(j)}, y_t, \mathbf{Y}_{t-1}) \\
&= \frac{p(s_t, \mathbf{S}_{t-1}^{(j)}, y_t, \mathbf{Y}_{t-1})}{p(\mathbf{S}_{t-1}^{(j)}, y_t, \mathbf{Y}_{t-1})} \\
&= \frac{p(y_t|s_t, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(s_t, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})}{p(y_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})} \\
&= \frac{p(y_t|s_t, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})}{p(y_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})} \\
&\propto p(y_t|s_t, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) \\
&\propto p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})p(s_t = a_i) \qquad (\text{See } Eq\ (4.23)) \\
p(s_t|\mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_t) &\propto \rho_{t,i}^{(j)}. \qquad (4.26)
\end{aligned}$$

From the state-space model of the system, it can be seen that the density $p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})$ is Gaussian and its mean and variance is calculated using the Kalman filtering algorithm,

$$p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) \sim N_c(\mu_t, \Sigma_t) \qquad (4.27)$$

The entire derivation for the application of the Kalman filtering algorithm to estimate the sampling density, $p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})$, is provided in the Appendix. For each $a_i \in A$, the *a posteriori* symbol probability, $p(s_t = a_i|\mathbf{Y}_t)$, can be estimated as [6,9],

$$\begin{aligned}
p(s_t = a_i|\mathbf{Y}_t) &= E\{\delta(s_t = a_i)|\mathbf{Y}_t\} \\
&\approx \frac{1}{W_t}\sum_{j=1}^{m}\delta(s_t = a_i)w_t^{(j)}, \qquad i = 1, ..., |A| \qquad (4.28)
\end{aligned}$$

where, $W_t = \sum_{j=1}^{m} w_t^{(j)}$. The decision on the symbol s_t is obtained as,

$$\begin{aligned}
\hat{s}_t &= \arg\max_{a_i \in A} p(s_t = a_i|\mathbf{Y}_t) \\
&\approx \arg\max_{a_i \in A} \sum_{j=1}^{m}\delta(s_t = a_i)w_t^{(j)} \qquad (4.29)
\end{aligned}$$

The particle filter algorithm for generating the sequential Monte Carlo samples of transmitted symbols $\{\mathbf{S}_t^{(j)}\}_{j=1}^m$ with corresponding importance weights $\{w_t^{(j)}\}_{j=1}^m$ and the Kalman filter update are given in the following algorithm as shown by Xiodong Wang, Rong Chen and Jun Liu [25]. Detailed derivations of all equations in the algorithm are provided in Appendix A.

C. Delayed Estimation

Since the fading process is highly correlated, the future received signals contain the information about current data and channel state. A delayed estimate is usually more accurate than the concurrent estimate. In delayed estimation [25], instead of making inference on $(\mathbf{x}_t, s_t)$ instantaneously with posterior distribution $p(\mathbf{x}_t, s_t|\mathbf{Y}_t)$, delay this inference to a later time $(t+\Delta)$, $\Delta > 0$, with the distribution $p(\mathbf{x}_t, s_t|\mathbf{Y}_{t+\Delta})$. There are two types of delayed estimation: the *delayed-weight* method [25] and the *delayed-sample* method [25].

Delayed-weight method: If the set $\{\mathbf{S}_t^{(j)}, w_t^{(j)}\}_{j=1}^m$ is properly weighted with respect to $p(\mathbf{S}_t|\mathbf{Y}_t)$, then by induction, the set $\{\mathbf{S}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)}\}_{j=1}^m$ is properly weighted with respect to $p(\mathbf{S}_{t+\delta}|\mathbf{Y}_{t+\delta})$. Hence, by focussing on $\mathbf{S}_t$ at time $(t+\delta)$, the delayed estimate of the symbol can be obtained as

$$p(s_t = a_i|Y_{t+\delta}) \approx \frac{1}{W_{t+\delta}} \sum_{j=1}^{m} \delta(s_t^{(j)} = a_i) w_{t+\delta}^{(j)}, \qquad i = 1, 2, ..., |A| \tag{4.30}$$

given in [25], where $W_{t+\delta} = \sum_{j=1}^m w_{t+\delta}^{(j)}$. Since the weights $\{w_{t+\delta}^{(j)}\}_{j=1}^m$ contain the information about the signals $(y_{t+1}, ..., y_{t+\delta})$, the estimate in Eq (4.30) is usually more accurate. However, this method requires some extra memory for storing $\{s_{t+1}^{(j)}, ..., s_{t+\delta}^{(j)}\}_{j=1}^m$.

Delayed-sample method: An alternative method is to generate both the delayed samples and the weights based on the signals $\mathbf{Y}_{t+\Delta}$, hence making the target distri-

Algorithm 6 Kalman-Particle Filter Algorithm

1: Kalman filter and importance weights are initialized as, $k_0^{(j)}=(\mu_0^{(j)}, \Sigma_0^{(j)})$, with $\mu_0^{(j)} = 0$ and $\Sigma_0^{(j)} = 2\Sigma, w_0^{(j)} = 1$, $j = 1, 2..., m$

2: One step predictive update of the Kalman filter,

$$\begin{aligned} K_t^{(j)} &= F\Sigma_{t-1}^{(j)}F^H + \mathbf{g}\mathbf{g}^H \\ \gamma_t^{(j)} &= \mathbf{h}^H K_t^{(j)}\mathbf{h} + \sigma^2 \\ \eta_t^{(j)} &= \mathbf{h}^H F\mu_{t-1}^{(j)} \end{aligned}$$

3: Compute the Trial sampling density, i.e., for each $a_i \in A$, we have,

$$p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) \sim N_c(a_i\eta_t^{(j)}, \gamma_t^{(j)})$$

4: Draw $s_t^{(j)}$ from the set A with probability,

$$p(s_t^{(j)} = a_i) \propto \rho_{t,i}^{(j)}, \qquad a_i \in A$$

Append $s_t^{(j)}$ to $\mathbf{S}_{t-1}^{(j)}$ and obtain $\mathbf{S}_t^{(j)}$

5: Compute the importance weights:

$$w_t^{(j)} \propto w_{t-1}^{(j)}.\sum_{a_i \in A}\rho_{t,i}^{(j)}$$

6: One step filtering update of Kalman filter,

$$\begin{aligned} \mu_t^{(j)} &= F\mu_{t-1}^{(j)} + \frac{1}{\gamma_t^{(j)}}(y_t - s_t^{(j)}\eta_t^{(j)})K_t^{(j)}\mathbf{h} \\ \Sigma_t^{(j)} &= K_t^{(j)} - \frac{1}{\gamma_t^{(j)}}K_t^{(j)}\mathbf{h}\mathbf{h}^H K_t^{(j)} \end{aligned}$$

7: For $j = 1, 2, ..., m$, retain $k_j = \lfloor w_t^{(j)} \rfloor$ copies of the sample set $(\mathbf{S}_t^{(j)}, k_t^{(j)})$.

8: Denote $K_r = m - \sum_{j=1}^{m} k_j$.

9: Obtain K_r i.i.d draws from the original sample set, with probabilities proportional to $(w_t^{(j)} - k_t^{(j)})$

10: Assign equal importance weights to all particles, $w_t^{(j)} = 1$.

bution at time $(t+\Delta)$. This procedure will provide better samples since it utilizes the future information in generating the current sample. But the algorithm is also more demanding both analytically and computationally because of the need of marginalizing out s_{t+d} for $d = 1, 2, ..., \Delta$. The dominant computation of the above delayed-sample method at each time involves the $(m|A|^{\Delta})$ one-step Kalman filter updates, which, as before, can be carried out in parallel.

D. Simulation Results

For the simulation, the following models namely ARMA(3,3) [18] and AR(2) [5] processes are used for the fading coefficients

$$\begin{aligned} \alpha_t &- 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3} \\ &= 10^{-2}(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} + 0.89409u_{t-3}) \end{aligned} \quad (4.31)$$

as given in [25], where, $u_t \sim N_c(0, 1)$

$$\alpha_t - 0.10\alpha_{t-1} - 0.80\alpha_{t-2} = u_t \quad (4.32)$$

where, $u_t \sim N_c(0, 0.27)$. The other parameters used for the simulation are:

- Modulation scheme = BPSK
- Number of particles N = 50,100
- Number of Monte Carlo runs = 100
- Number of transmitted symbols = 100000
- For delayed estimation, delay = 2 is considered.
- For the case of non-Gaussian noise $\epsilon = 0.1$ and $k = 10$.

Steps carried out for the simulation of the particle filtering algorithm for the system:

1. Obtain the randomly generated BPSK signals and differentially encode them before transmission.

2. Generate the true states and the observations using Eq (4.9).

3. Generate sequential Monte Carlo samples of transmitted symbols $\{\mathbf{S}_t^{(j)}\}_{j=1}^{m}$ with corresponding importance weights $\{w_t^{(j)}\}_{j=1}^{m}$ at time t by using the Algorithm 6.

4. Calculate the effective sample size $\overline{m}_t$.

5. If $\overline{m}_t \leq m/10$, then resample else go back to Step 3.

6. For each $a_i \in A$, the *a posteriori* symbol probability $p(s_t = a_i|\mathbf{Y}_t)$ is calculated.

7. The symbol is decoded and the bit error rate (BER) is calculated between transmitted symbols and decoded symbols.

Steps from 1 to 7 are repeated for each independent Monte Carlo run and BER is averaged over all Monte Carlo runs. For comparison, known channel bound (MLSE) and performance of differential detector is also plotted.

Fig. 11 shows BER performance of the system with fading coefficients modeled as ARMA process in the presence of additive Gaussian noise. It is evident that the delayed weighted method gives better performance, for instance at SNR of 30 dB it gives BER 0.0014 while the particle filter with zero delay gives BER of 0.0027. For comparison, the performance using differential detection method is also plotted and it is seen to perform poorly especially from SNR of 20 dB-40dB and saturates at a BER value of 0.0101. Besides that, known channel bound is also plotted for comparison. It

is seen that the delayed weighted scheme performance is close to the known channel bound.

Fig. 12 shows BER performance of the system with fading coefficients modeled as ARMA process in an additive non-Gaussian noise. Differential detection forms error floor from SNR of 30dB-40dB and saturates at 0.0115. It is seen that from SNR of 10dB-20dB the delayed weight method and particle filter with zero delay are close in performance to the known channel bound. At higher values of SNR, typically from 25dB- 40dB, delayed weighted method shows a large performance improvement when compared to the differential detection method.

Fig. 13 shows BER performance of the system with fading coefficients modeled as AR process in additive Gaussian noise. At lower values of SNR, typically from 10dB-15dB, differential detector, particle filter with zero delay and delayed weighted method are close in the performance. Delayed weight method and particle filter with zero delay shows much improvement than differential detector from SNR of 25dB-40dB. Delayed weight method gives a BER of 0.0005 at SNR of 40dB.

Fig. 14 shows BER performance of the system with fading coefficients modeled as AR process and additive non-Gaussian noise. Here also at lower SNR (i.e., from 10dB-15dB) values differential detector, delayed weight method, particle filter with zero delay performs closely. Delayed weight method shows a close performance to known channel bound from SNR of 20-25 dB. For instance at 25dB, delayed weight method gives BER of 0.0031 while known channel bound gives BER of 0.0024. By increasing the delay, the delayed weight method performs close to the optimal detector.

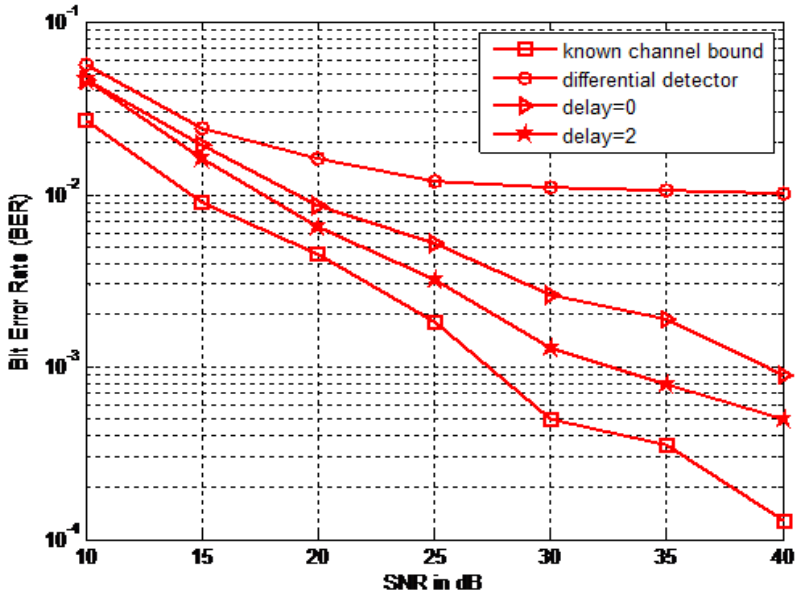

Fig. 11. BER with ARMA fading model in the presence of additive Gaussian noise

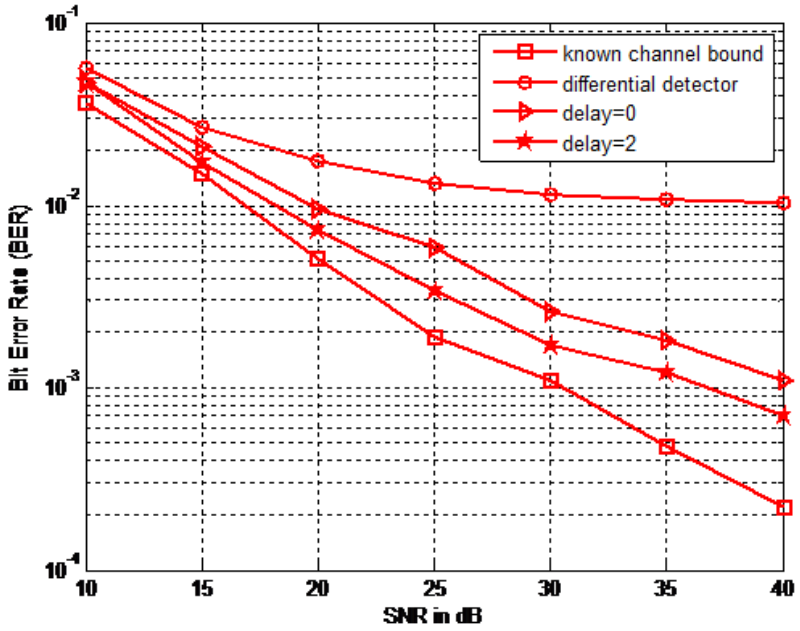

Fig. 12. BER with ARMA fading model in the presence of additive non-Gaussian noise

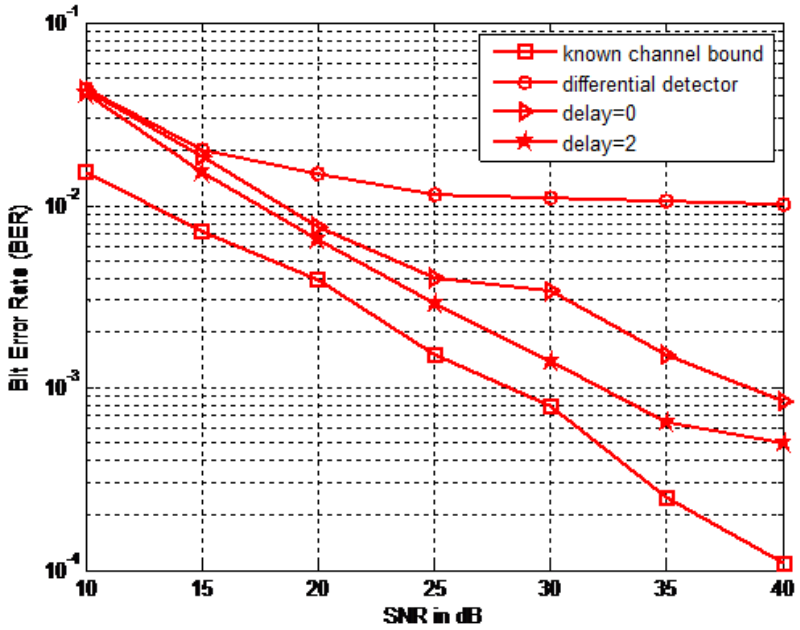

Fig. 13. BER with AR fading model in the presence of additive Gaussian noise

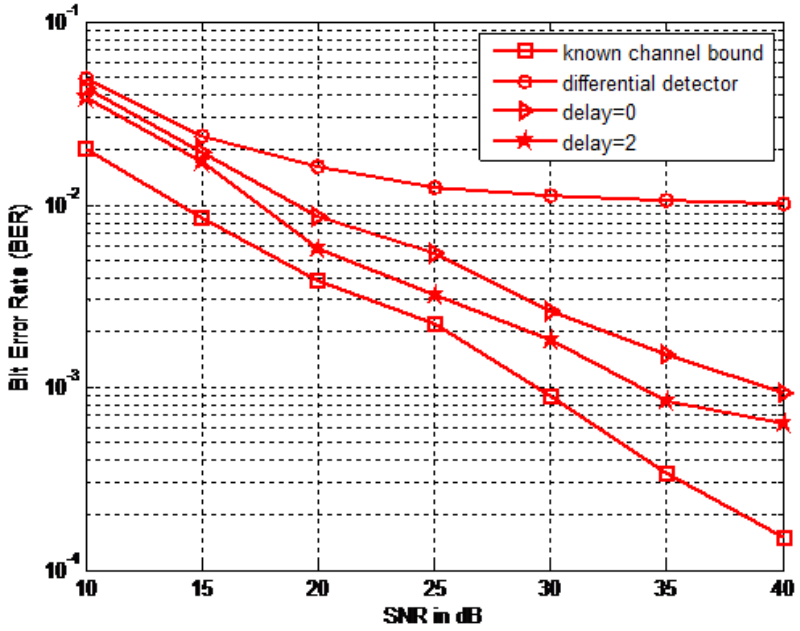

Fig. 14. BER with AR fading model in the presence of additive non-Gaussian noise

CHAPTER V

CALCULATION OF CAPACITY USING PARTICLE FILTER

The information carrying capacity of a communication channel was first considered by Shannon in 1948 [27] who calculated the capacity of a memoryless channel with additive white Gaussian noise (AWGN) for a given signal-to-noise ratio (SNR). The *capacity* of a channel [27,28] can be defined to be the tightest upper bound on the amount of information that can be transmitted over a communications channel with arbitrarily small probability of error. To determine whether a redundant source can be communicated through a noisy channel, we make the following assumptions. A source symbol is generated every T_s seconds and an optimal source code, whose average code length per source symbol equals the entropy rate, is used. For a discrete memoryless source S, the entropy rate (in bits) is defined to be

$$H(S) = E[-\log_2(p_S)]. \tag{5.1}$$

If the entropy rate of the source is written as $H(S)$, the channel encoder will receive on average $H(S)/T_s$ information bits per second from the source. Assume that a code symbol is input to the channel every T_c seconds. The channel capacity C is the maximum rate (in information bits per channel symbol) that can be transmitted over the chanel reliably. In order to transmit all the information from the source, the channel must be able to transmit

$$C > R = \frac{H(S)T_c}{T_s} \tag{5.2}$$

information bits per channel symbol, where R is the information rate (in bits per channel symbol) of the channel encoder [27]. By transmitting information with rate R, the channel is used every T_c seconds. It is common to represent the channel capacity

within a unit bandwith of the channel and is measured typically in bits/s/Hz.

A. Capacity - Definition

The channel capacity is given by the maximum of the mutual information between the input and output of the channel, where the maximization is with respect to the input distribution [28]. If the input and output of a memoryless wireless channel are represented with the random variables X and Y respectively, the channel capacity is defined as,

$$C = \max_{p(x)} I(X;Y) \tag{5.3}$$

where $I(X;Y)$ represents the mutual information between X and Y. Mutual information [27,28] is a measure of the amount of information that one random variable contains about another variable. Mathematically, it is given by

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) && (5.4) \\ &= H(X) - H(X|Y) && (5.5) \end{aligned}$$

where $H(Y|X)$ is the conditional entropy between the random variables X and Y. The entropy of a random variable can be described as a measure of the uncertainty of the random variable.

B. Capacity Calculation - Basic Method

The problem of computing the information rate between an input process $X = (X_1, ..., X_n)$ and an output process $Y = (Y_1, ..., Y_n)$ of a time-invariant discrete-time channel with memory is discussed by Dauwels and Loeliger [14]. There were many methods developed previously by various researchers when the input alphabet and the state space are finite [29,30,31,32]. But Dauwels and Loeliger [14] extended this result

to a continuous state space. They used the particle filtering approach to achieve this. But before going into the details of the particle filtering method, a brief introduction of the basic method [29] for computing the capacity, as presented in [14], is given below.

Let $x_k^n \triangleq (x_k, x_{k+1}, ..., x_n)$ and $x^n \triangleq (x_1, x_2, ..., x_n)$. For a stationary ergodic sequence of random variables, the following results hold,

- The sequence, $-\frac{1}{n}\log p(X^n) \overset{a.s}{\to} H(X)$
- The sequence, $-\frac{1}{n}\log p(Y^n) \overset{a.s}{\to} H(Y)$
- The sequence, $-\frac{1}{n}\log p(X^n, Y^n) \overset{a.s}{\to} H(X) + H(Y|X)$

Using these results, the value of $I(X;Y)$ given by Eq (5.4) can be calculated as below:

Algorithm 7 Basic Method

1: Sample two very long sequences x^n and y^n jointly from the distribution $p(x^n, y^n)$.
2: Compute $\log p(x^n)$, $\log p(y^n)$, and $\log p(x^n, y^n)$. If $H(Y|X)$ is known analytically, then the value of $\log p(y^n)$ is sufficient.
3: Use the values computed in previous step to calculate the estimate, $\hat{I}(X;Y) \triangleq \frac{1}{n}\log p(x^n, y^n) - \frac{1}{n}\log p(x^n) - \frac{1}{n}\log p(y^n)$

If the state space is finite, the values in the above method can be obtained by using forward sum-product recursion of the BCJR algorithm [33]. Using this algorithm, we have from [33],

$$p(y^n) = \int_{x^n}\int_{s_0^n} p(x^n, y^n, s_0^n) dx^n ds_0^n \tag{5.6}$$

Let $\mu_k(s_k) \triangleq p(s_k|y^k)$. Then using the algorithm in [33] we have,

$$\mu_k(s_k) \sim \int_{x^k} \int_{s_0^{k-1}} p(x^k, y^k, s_0^k) dx^k ds_0^{k-1} \tag{5.7}$$

for k = 1,2,3... with $\mu_0(s_0) \triangleq p(s_0)$.

But the state metrics tend to zero quickly and to counter this the recursive process is varied so that,

$$\mu_k(s_k) = \lambda_k \int_{x_k} \int_{s_{k-1}} \mu_{k-1}(s_{k-1}) p(x_k, y_k, s_k|s_{k-1}) dx_k ds_{k-1} \tag{5.8}$$

where $\lambda_1, \lambda_2, ...$ are positive scalar factors obtained by setting $\int_{s_k} \mu_k(s_k) ds_k = 1$. Then the value of the estimate will be given by

$$\frac{1}{n} \sum_{k=1}^{n} \log \lambda_k = -\frac{1}{n} \log p(y^n) = H(Y) \tag{5.9}$$

The other values of $\log p(x^n)$ and $\log p(x^n, y^n)$ can also be calculated in the same manner.

C. Particle Method

The computation of the integrals in the Eq (5.6) becomes tractable and practical only if the input alphabet and the state space is finite. But if the state space or the input alphabet is not finite, then the above method becomes impractical. This problem is overcome by using the particle filtering approach as shown by Dauwels and Loeliger [14]. This section provides some details of their work.

The particle filtering approach can be understood as a message passing algorithm where the probability distributions are represented by particles. The set of N particles is represented as $\{\hat{x}^{(l)}, w^{(l)}\}_{l=1}^{N}$, where $\hat{x}^{(l)} \in \chi$ and χ is the input alphabet, the $w^{(l)}$ are positive particle weights such that $\sum_{l=1}^{N} w^{(l)} = 1$. Extending this theory to the

present problem, the distribution $p(s_{k-1}, s_k, x_k|y^{k-1})$ is represented by a list of N weighted particles, $\{(\hat{s}_{k-1}^{(l)}, \hat{s}_k^{(l)}, \hat{x}_k^{(l)}), w_{k-1}^{(l)}\}_{l=1}^N$. Then the value of λ_k can be obtained by

$$\lambda_k^{-1} = \sum_{l=1}^{N} w_{k-1}^{(l)} p(y_k|\hat{s}_{k-1}^{(l)}, \hat{s}_k^{(l)}, \hat{x}_k^{(l)}) \tag{5.10}$$

The method described above can be shown in as an algorithm [14],

Algorithm 8 Particle Method

1: Begin with a particle list $\{(\hat{s}_{k-1}^{(l)}, \hat{s}_k^{(l)}, \hat{x}_k^{(l)}), w_{k-1}^{(l)}\}_{l=1}^N$ that represents μ_{k-1}. This set can be obtained by sampling from $p(x_k, s_k|s_{k-1})$.

2: Compute the value of estimate of λ_k using Eq (5.10).

3: Calculate the weights using $w_k^{(l)} = \lambda_k w_{k-1}^{(l)} p(y_k|\hat{s}_{k-1}^{(l)}, \hat{s}_k^{(l)}, \hat{x}_k^{(l)})$

4: Using the values computed in Steps 2 and 3, update the particle list to represent μ_k.

5: Perform any one of the resampling techniques to update the particle list $\{\hat{s}_k^{(l)}, w_{k-1}^{(l)}\}_{l=1}^N$.

D. Application to a Flat-Fading Channel

This section discusses the particle filtering approach for computing the capacity [14] of the wireless flat-fading channel introduced in the previous chapter. Note the difference in the notation of the model as $\{x_t\}$ here represents the state variables and $\{s_t\}$ represents the input symbols. This is to maintain the same notation as in the previous chapter where this model is introduced.

The particle method described in the previous section can be utilized here as the state sequence in this particular channel estimation problem is not finite. Now,

consider the state space equation of the flat-fading channel introduced in the previous chapter,

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{5.11}$$

$$y_t = s_t\mathbf{h}^H\mathbf{x}_t + \sigma v_t \tag{5.12}$$

where $\mathbf{x}_t \triangleq [x_t \ x_{t-1} \ x_{t-2} \ ... \ x_{t-r}]^T$ is the state variable corresponding to the fading coefficients α_t, y_t represents the received signal, and $\{v_t\}$ is a white complex Gaussian noise sequence with unit variance and independent real and imaginary components. Let BPSK modulation scheme be used for the transmission over this channel.

Now, the particle filtering algorithm for the calculation of capacity, which is discussed in the previous section, will be applied for this case. The algorithm is slightly modified for getting the maximum accuracy in the estimate of the capacity. Here, the particles are assumed to be mean and variance of the estimate instead of assuming a point mass estimate. Using the mean and variances as particles, a more accurate estimate can be obtained using a smaller number of particles as compared to that of a point mass estimation. This improvement may also be understood intuitively, by using the representation of mean and variance of the estimate as particles the entire probability space of the estimate can be fully covered using a smaller number of samples than using a point mass particles to cover the space. This method of particle filtering is already discussed in the previous chapter for the detection of the symbols over the same adaptive channel. This particular algorithm is now extended and modified so that the capacity of that channel can be calculated. For the simulation, the ARMA(3,3) [18] processes are used for the fading coefficients [25],

$$\begin{aligned} \alpha_t &- 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3} \\ &= 10^{-2}(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} + 0.89409u_{t-3}) \end{aligned} \tag{5.13}$$

where, $u_t \sim N_c(0,1)$.

The other parameters used for the simulation are:

- Modulation scheme = BPSK
- Number of particles N = 50
- Number of transmitted symbols = 10000

Steps carried out for the simulation of the particle filtering algorithm for the system:

1. Obtain the randomly generated BPSK signals and differentially encode them before transmission.
2. Generate the true states and the observations using Eq (5.11).
3. Generate sequential Monte Carlo samples of transmitted symbols $\{\mathbf{S}_t^{(j)}\}_{j=1}^m$ with corresponding importance weights $\{w_t^{(j)}\}_{j=1}^m$ at time t by using the Algorithm 6 (Chapter IV).
4. Calculate the conditional probabilities, $p(y_k|y_{k-1})$ and $p(y_k|s_k, y_{k-1})$ based on the estimated symbol at each time k for all the particles
5. Calculate the effective sample size $\overline{m}_t$.
6. If $\overline{m}_t \leq m/10$, then resample the particles along with both the entropies else go back to Step 3.
7. Then the estimate of the conditional probabilities at each time is calculated based on the weights and probabilities for all particles.
8. Now the entropy value, $H(Y) = -E\{\ln p(y_k|y_{k-1})\}$ is calculated by averaging over the log probabilities at all the time instants.

9. Now the entropy value, $H(Y|S) = -E\{\ln p(y_k|s_k, y_{k-1})\}$ is calculated by averaging over the log probabilities at all the time instants.

10. Finally the capacity of the channel is calculated using the entropies, $I(S;Y) = H(Y) - H(Y|S)$.

The following figures show the results of applying the particle filtering algorithm.

Fig. 15 shows the variation of estimated information rate of the adaptive wireless channel with fading coefficients modeled as ARMA process in the presence of additive Gaussian noise with SNR in dB. It also shows the plot for various values of the process noise variance. It can be clearly seen that the information rate increases with SNR as which is expected for any communication channel. Besides that, the value of the information rate can be seen to improve with the decrease in the value of the process noise variance.

Fig. 16 shows the estimated information rate as a function of the sequence length, i.e., the number of symbols transmitted over the channel, for 10 simulation runs of the particle filtering method. It is clear from the figure that the particle filtering algorithm settles down to the exact value as the number of symbols is increased. It means that for a very large input data, the estimate provides the exact capacity of the channel.

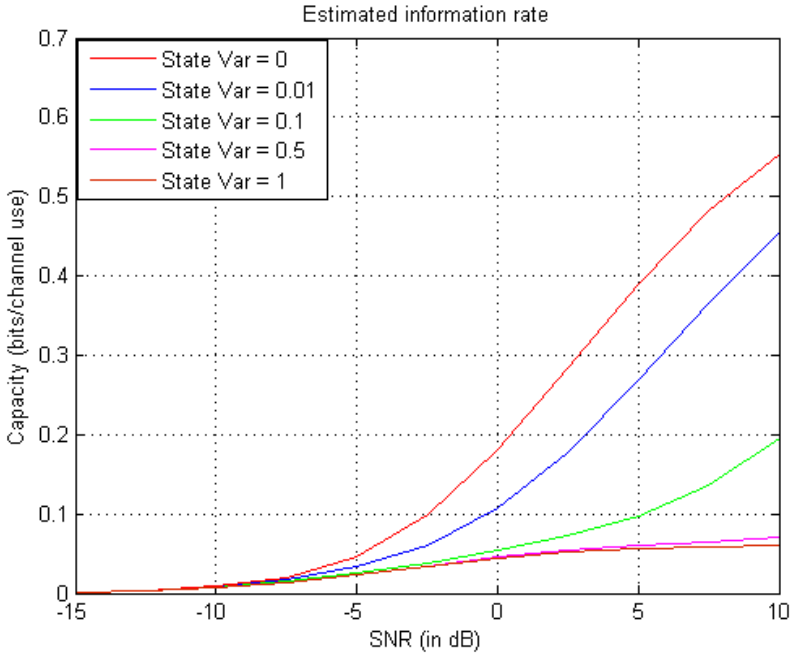

Fig. 15. Variation of estimated capacity of the channel with SNR (in dB)

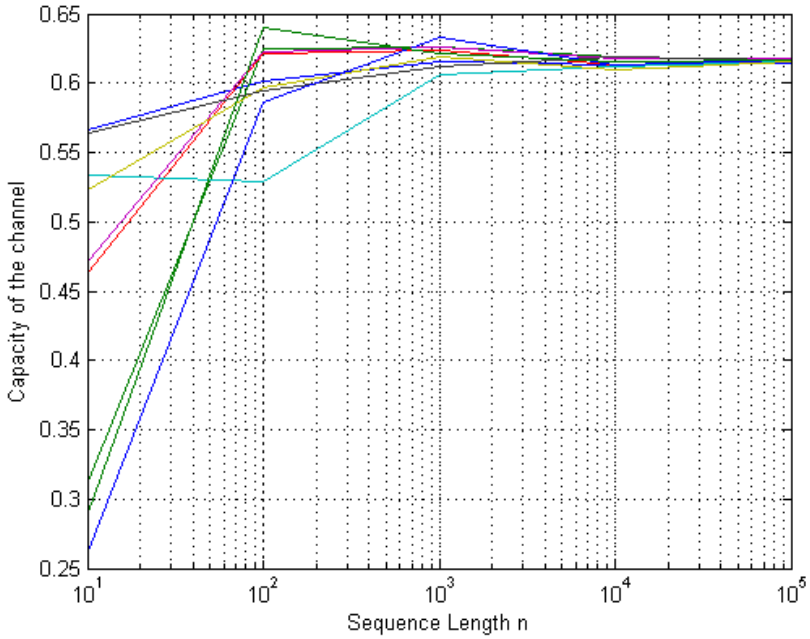

Fig. 16. Estimated information rate as a function of the sequence length n, for 10 simulation runs of the particle method SNR = 10dB

CHAPTER VI

PARTICLE FILTERING OF CONTINUOUS-DISCRETE SYSTEMS

This chapter discusses the application of particle filtering to continuous-discrete optimal filtering problems, where the system model is of the form of a stochastic differential equation (SDE), and the noisy observations of the system are obtained at discrete instances of time. The Girsanov theorem, which is used for evaluating the likelihood ratios needed in importance sampling, is explained. Rao-Blackwellization of conditionally Gaussian models and unknown static parameter models is also considered at the end of the chapter. An example of estimating the angular acceleration of a simple pendulum, dealt by Simo Sarkka [15], is considered and simulation results are provided.

A. Optimal Filtering

Optimal filtering is defined as a filtering method that can be used for estimating the states of time varying systems using a set of noisy measurements. The state of the system may refer to the dynamic variables such as position, velocities and accelerations or orientation and rotational motion parameters, which describe the physical state of the system [15]. The noise in the measurements refers to a noise in the sense that the measurements are uncertain. The time evolution of the state is modeled as a dynamic system, which is perturbed by a certain process noise. This noise is used for modeling the uncertainties in the system dynamics.

These type of problems can be seen very often in many engineering applications. These kind of models can be found, for example, in navigation, aerospace engineering, space engineering, remote surveillance, telecommunications, physics, audio signal processing, control engineering, finance and several other fields. All these applications

can be modeled as discrete-time, continuous-discrete or continuous-time optimal filtering problems, depending on whether the state and measurements are discrete or continuous functions of time [15,34].

1. Optimal Continuous-Discrete Filtering

Most of the physical systems are often modeled as continuous-discrete as in the Nature time is continuous (Jazwinski [35]). In continuous-discrete filtering, the state dynamics are modeled as continuous-time stochastic processes and the measurements are obtained at discrete instances of time. The idea of this type of filtering can be understood by considering a time series, which is not measured on each time step but instead between the discrete time steps (measurement steps) there are additional states as shown in the Fig 17. If an infinite number of additional states are assumed between the measurement steps, the state sequence becomes a random function, which is observed at discrete instances of time.

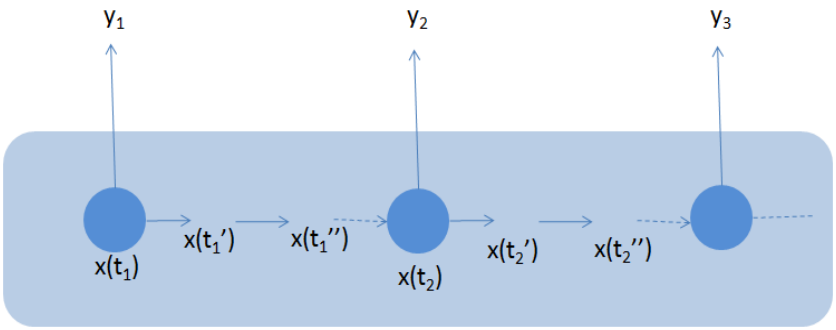

Fig. 17. Continuous-discrete filtering as a limiting case of discrete time filtering

The state dynamics are mathematically modeled as a stochastic differential equation (SDE) [36,37], which can be defined as ordinary differential equations driven by

random white noise processes $w(t)$ as [15],

$$\frac{dx}{dt} = f(x,t) + L(t)w(t) \tag{6.1}$$

Here $x(t)$ is the state, $f(x,t)$ is the drift function, $L(t)$ is the dispersion matrix, and $w(t)$ is the white noise process. In continuous-discrete filtering the measurements y_k are obtained at discrete time instances $\{t_1, t_2, ...\}$. The measurement model is of the same form as in the case of other problems that are considered in the previous chapters. However, the advantage of this model formulation over the discrete model formulation is that the time step size $t_k = t_{k+1} - t_k$ does not need to be constant. As discussed in the previous chapters, the optimal solution to this type of filtering problem can be computed by the following prediction and update steps [35]:

- **Prediction step** solves the predicted probability density at time t_k from the Kolmogorov forward partial differential equation using the old posterior probability density at time t_{k-1} as the boundary condition.

- **Update step** uses the Bayes rule for computing the posterior probability density of state at time t_k from the probability density obtained in the prediction step, and the measurement y_k.

A smoothing step may be used following the filtering step so that a more accurate estimate. In the smoothing step, the past values of the estimate are updated conditioned on the past and present values of the measurements. The distributions for the prediction, update and smoothing stages are given below:

- **Filtering distribution** of the state $x(t_k)$ at the time t_k given the measurement y_k is given by:

$$p(x(t_k)|y_1, ..., y_k) \tag{6.2}$$

- **Prediction distributions** can be computed for all $t > t_k$,

$$p(x(t)|y_1, ..., y_k), \qquad t > t_k \tag{6.3}$$

by solving the corresponding Kolmogorov forward equation. The marginal posterior distributions of the states between the measurements conditional to all the previous measurements are given by these prediction distributions.

- **Smoothing distributions** can be computed for all times $t \in [0, t_T]$ if the measurements up to y_T:

$$p(x(t)|y_1, ..., y_T), \qquad 0 < t < t_T \tag{6.4}$$

The dynamics of the processes are modeled as *Ito stochastic differential equations* (SDE) driven by Brownian motions and the measurements are modeled as non-linear functions of the state, which are corrupted by Gaussian measurement noises. In the next section, the various measure transformation based methods for continuous-discrete sequential importance resampling are presented.

2. Continuous-Discrete Sequential Importance Resampling

The methods used in this type of filtering are based on transformations of probability measures by the *Girsanov* theorem [35-38], which is a theorem from mathematical probability theory. This theorem can be used for calculating the likelihood ratios of stochastic processes. It states that the likelihood ratio of a stochastic process and Brownian motion, that is, the Radon-Nikodym derivative of the measure of the stochastic process with respect to the measure of Brownian motion, can be represented as an exponential martingale which is the solution to a certain stochastic differential equation. This theorem will fit in the present problem under consideration as the

state-space is modeled as a stochastic differential equation.

The state space model for these problems can be generalized using Eq (6.1),

$$\begin{aligned} dx &= f(x,t)dt + L(t)d\beta(t) \\ y_k &\sim p(y_k|x(t_k)) \end{aligned} \tag{6.5}$$

where $\beta(t)$ is a Brownian motion with positive definite diffusion matrix $Q_c(t)$, $L(t)$ is an invertible matrix called dispersion matrix for all $t \geq 0$ and the initial conditions are $x(0) \sim p(x(0))$. The purpose of the Bayesian optimal continuous-discrete filter is to compute the posterior distribution of the current state $x(t_k)$ given the measurements up to the current time. Further it is assumed that an importance process $s(t)$, which is defined by a SDE, exits and given by:

$$ds = g(s,t)dt + B(t)d\beta(t) \tag{6.6}$$

where the matrix $B(t)$ is invertible for all t.

The process $s(t)$ is an approximation to the optimal result and so using it as the importance process will produce more accurate presentation of the filtering distribution. This is possible as a less degenerate particle set will be produced. Because the matrices $L(t)$ and $B(t)$ are assumed to be invertible, the probability measures of x and s are absolutely continuous and thus satisfies the condition for applying the Girsanov theorem [34]. An algorithm showing the steps for sequential importance sampling to continuous discrete (CD-SIR) dynamic models, as shown in [34], is given below. The SIR algorithm recursion starts by drawing samples $\{x_0^{(i)}\}$ from the initial distribution and setting $w_0^{(i)} = \frac{1}{N}$, where N is the number of Monte Carlo samples.

The importance process in the above algorithm can be obtained by using any

Algorithm 9 Continuous Discrete SIR Algorithm

1: **procedure** CD-SIR($s(t), x_{k-1}^{(i)}, w_{k-1}^{(i)}, y_k$)

2: Simulate N realizations of the importance processes from $t = t_{k-1} : t_k$

$$\begin{aligned} ds^{(i)} &= g(s^{(i)}, t)dt + B(t)d\beta^{(i)}(t), \qquad s^{(i)}(t_{k-1}) = x_{k-1}^{(i)} \\ ds^{*(i)} &= L(t)B^{-1}(t)ds^{(i)}(t), \qquad s^{(i)}(t_{k-1}) = x_{k-1}^{(i)} \end{aligned}$$

where $\beta^{(i)}(t)$ are independent Brownian motions and $i = 1, 2, ..., N$

3: Now, calculate the log-likelihood ratios from $t = t_{k-1} : t_k$

$$\begin{aligned} d\Lambda^{(i)} &= \{f(s^{*(i)}(t), t) - L(t)B^{-1}(t)g(s^{(i)}(t), t)\}^T \\ &\quad \times L^{-T}(t)Q^{-1}(t)d\beta^{(i)}(t) \\ &\quad -\frac{1}{2}\{f(s^{*(i)}(t), t) - L(t)B^{-1}(t)g(s^{(i)}(t), t)\}^T \\ &\quad \times \{L(t)Q(t)L^T(t)\}^{-1} \\ &\quad \times \{f(s^{*(i)}(t), t) - L(t)B^{-1}(t)g(s^{(i)}(t), t)\}dt \end{aligned} \tag{6.7}$$

where $\Lambda^{(i)}(t_{k-1}) = 0$ and set $x_k^{(i)} = s^{*(i)}(t_k)$, and $Z_k^{(i)} = \exp\{\Lambda^{(i)}(t_k)\}$

4: For each i, calculate the importance weights and normalize to unity

$$w_k^{(i)} = w_{k-1}^{(i)} Z_k^{(i)} p(y_k | x_k^{(i)}) \tag{6.8}$$

5: Based on the effective number of particles, the resampling is performed.

6: **end procedure**

of the filtering techniques available for non-linear cases, for example, the extended Kalman filter (EKF). Simo Sarkka has discussed some of the numerical methods in detail in his doctoral thesis work [15]. His work may be referred for further details into the simulation of this algorithm.

B. Rao-Blackwellization of Models with Static Parameters

Now a dynamic model with unknown static parameters is considered. This kind of models can be handled such that only the inner process is sampled and the linear part is integrated out using the continuous-discrete Kalman filter. Then it is possible to form a Rao-Blackwellized estimate, where the probability density is approximated by a mixture of Gaussian distributions.

In generic particle filtering, Rao-Blackwellization (RB) refers to a filtering method of integrating out a part of the state analytically [39]. This method will result in a drastic reduction in the variance of the particles. The main advantage of this method is that, for the same performance level, fewer samples will be needed.

If the posterior distribution of the unknown static parameter θ depends only on a set of sufficient statistics $T_k = T_k(x(t_1), ..., x(t_k), y_{1:k})$, then it can be marginalized out and only the state needs to be sampled. The state space equations for such a dynamic model can be given by

$$\begin{aligned} dx &= f(x, t, \theta)dt + L(t, \theta)d\beta(t) \\ y_k &\sim p(y_k|x(t_k), \theta) \end{aligned} \tag{6.9}$$

Now assume that the prior distribution of θ is given by

$$p(\theta) = p(\theta|T_0) \tag{6.10}$$

Also, assume that the conditional posterior will follow,

$$p(\theta|x(t_1),...,x(t_k),y_{1:k}) = p(\theta|T_k) \tag{6.11}$$

$$T_k = \phi(T_{k-1},x(t_k),y_k) \tag{6.12}$$

Thus, the marginal likelihood equation can be written as follows:

$$p(y_k|x(t_k),T_{k-1}) = \int p(y_k|x(t_k),\theta)p(\theta|T_{k-1})d\theta \tag{6.13}$$

An algorithm for this RBPF method can be given as [34]:

Algorithm 10 Continuous Discrete RB SIR Algorithm

1: **procedure** CDRB-SIR($x_{k-1}^{(i)}, T_{k-1}^{(i)}, w_{k-1}^{(i)}, y_k$)

2: Perform steps 1 and 2 of Algorithm 9 discussed in the previous section, and calculate the importance process and likelihood ratio

3: Obtain the values of $x_k^{(i)}$ and $Z_k^{(i)}$ as given in Algorithm 9

4: For each i, calculate the following,

$$T_k^{(i)} = \phi(T_{k-1}^{(i)}, x_k^{(i)}, y_k)$$

5: Now, calculate the importance weights according to the following equation and normalize them to unity

$$w_k^{(i)} = w_{k-1}^{(i)} Z_k^{(i)} p(y_k|x_k^{(i)}, T_{k-1}^{(i)})$$

6: Based on the effective number of particles, the resampling is performed.

7: **end procedure**

C. Simulation example

In this section the continuous-discrete sequential importance sampling (CD-SIR) is applied to estimate the angular position of a simple pendulum which is distorted by a random noise term given some partial observations on the position. This example is dealt by many researchers as it forms a perfect example to showcase the efficiency of a CD-SIR. The main reference for this section is from the doctoral thesis of Simo Sarkka [15].

The dynamic model for the angular position of simple pendulum is driven by a stochastic differential equation (SDE), which is distorted by random white noise accelerations $w(t)$ with the spectral density q. This model is given by

$$\frac{d^2x}{dt^2} + a^2 \sin(x) = w(t) \tag{6.14}$$

where a is the angular velocity of the pendulum.

Now, let $\mathbf{x} = [x_1 \;\; x_2]^T = [x \;\; dx/dt]^T$. Then, the state space form can be changed to the following model in terms of Brownian motion $\beta(t)$ which has a diffusion coefficient of q,

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ dx_2 &= -a^2 \sin(x_1) + d\beta \end{aligned} \tag{6.15}$$

Let the state of the pendulum be measured once per unit time and assume that the measurements are corrupted by Gaussian noise with an unknown variance σ^2. Then the measurement model can be given as,

$$\begin{aligned} y_k &\sim N(x_1(t_k), \sigma^2) \\ \sigma^2 &\sim Inv - \chi^2(v_0, \sigma_0^2) \end{aligned} \tag{6.16}$$

The variance σ^2 is now an unknown static variable, where the procedure of Rao-Blackwellization that is discussed in the previous section can be applied. Based on the posterior distribution of σ^2 given in Eq (6.16) and given the state of the system, $x(t_k)$, the marginal distribution of the measurement at that step k is given by

$$\begin{aligned} p(y_k|x(t_k)) &= \int N(y_k|x_1(t_k),\sigma^2)Inv-\chi^2(\sigma^2|v_{k-1},\sigma^2_{k-1})d\sigma^2 \\ &= t_{v_k}(y_k|x_1(t_k),\sigma^2_k) \end{aligned} \tag{6.17}$$

with parameters,

$$\begin{aligned} v_k &= v_{k-1}+1 \\ \sigma^2_k &= \frac{v_{k-1}\sigma^2_{k-1}+(y_k-x_1(t_k))^2}{v_k} \end{aligned} \tag{6.18}$$

Here, t_{v_k} represents the Student's T distribution. Now, the importance process can be formed by using either EKF or UKF such that a Gaussian approximation to the posterior distribution of the state $\mathbf{x}(t_k) = [x_1(t_k)\ \ x_2(t_k)]^T$ is obtained. This approximation needs that the variance σ^2 is assumed to be known for which a distribution is already known from Eq (6.18).

The set of particles at time step $k-1$ comprises of $\{w^{(i)}_{k-1}, x^{(i)}_{1,k-1}, x^{(i)}_{2,k-1}, v^{(i)}_{k-1}, \sigma^{2,(i)}_{k-1}\}$. The resulting particle filtering algorithm for this problem can be shown as below [31]:

D. Simulation Results

This section provides the plots for the simple pendulum problem discussed previously. The EKF, UKF estimates are also computed for the problem so as to be able to compare the results with that of particle filtering method. The following parameters are used for the simulation:

Algorithm 11 RBPF - Simple pendulum with Noise

1: For each particle, perform EKF/UKF prediction from t_{k-1} to t_k and update given the measurement y_k. Assume that the marginal mean and covariance of $x_2(t_k)$ are $m_{2,k}^{(i)}$ and $P_{22,k}^{(i)}$.

2: Initialize the scaled important process $s_1^{*(i)}(t_{k-1}) = x_{1,k-1}^{(i)}$, $s_2^{*(i)}(t_{k-1}) = x_{2,k-1}^{(i)}$ and $\lambda^{*(i)}(t_{k-1}) = 0$.

3: For each i, simulate the scaled importance process, and the logarithm of likelihood ratio from time t_{k-1} to time t_k [15]:

$$\begin{aligned}
\frac{ds_1^{*(i)}}{dt} &= s_2^{*(i)} \\
ds_2^{*(i)} &= \left(\frac{m_{2,k}^{(i)} - x_{2,k-1}^{(i)}}{\sqrt{P_{22,k}^{(i)} \Delta t}} \right) dt + q^{1/2} d\beta \\
d\lambda^{*(i)} &= -\frac{a^2}{q^{1/2}} \sin(s_1^{*(i)}(t)) d\beta - \left(\frac{m_{2,k}^{(i)} - x_{2,k-1}^{(i)}}{\sqrt{P_{22,k}^{(i)} \Delta t}} \right) d\beta \\
&- \frac{a^2}{q^{1/2}} \sin(s_1^{*(i)}(t)) \left(\frac{m_{2,k}^{(i)} - x_{2,k-1}^{(i)}}{\sqrt{P_{22,k}^{(i)} \Delta t}} \right) dt \\
&- \frac{a^4}{q} \sin^2(s_1^{*(i)}(t)) dt - \frac{1}{2} \left(\frac{m_{2,k}^{(i)} - x_{2,k-1}^{(i)}}{\sqrt{P_{22,k}^{(i)} \Delta t}} \right)^2 dt
\end{aligned} \tag{6.19}$$

4: Now, calculate the new sufficient statistics for the variance, given by

$$\begin{aligned}
v_k^{(i)} &= v_{k-1}^{(i)} + 1 \\
\sigma_k^{2,(i)} &= \frac{v_{k-1}^{(i)} \sigma_{k-1}^{2,(i)} + (y_k - x_{1,k}^{(i)}))^2}{v_k^{(i)}}
\end{aligned} \tag{6.20}$$

5: Now, calculate the importance weights according to the following equation and normalize them to unity

$$w_k^{(i)} = w_{k-1}^{(i)} Z_k^{(i)} t_{v_k^{(i)}}(y_k | x_{1,k}^{(i)}, \sigma_k^{2,(i)}) \tag{6.21}$$

6: Based on the effective number of particles, the resampling is performed.

- Number of particles m = 1000, 10000
- Process noise spectral density, q = 0.01
- Angular velocity, a = 1
- Sampling step size = 0.1
- True measurement variance, $\sigma^2 = 0.25$
- Prior distribution for the unknown variance, $\sigma^2 \sim Inv - \chi^2(2, 0.2)$

The particle filtering algorithm described in the previous section is run with the above parameters. The results are provided below. Fig. 18 to Fig. 21 show the results with the use of extended Kalman filtering and unscented Kalman filtering. It also shows the effect of using a smoothing distribution which is just a backward filter that calculates the state based on the present and past measurements. The details for realization and usage of a smoothing filter in the context of continuous discrete filtering can be obtained from [15]. Fig. 22 to Fig. 25 show the results with the use of particle filtering algorithm for varying number of particles. It clearly shows the improvement with increase in the number of particles used. It also shows the effect of a smoothing filter at the end of the estimate.

Table II shows the values of RMSE for all the filtering techniques used. Apparently, the performance is better for the particle filtering algorithm with a smoothing distribution. Also as the number of particles increases the estimate becomes more accurate and hence the error comes down.

Table II. RMSE comparison for the noisy simple pendulum problem

S.No.	**Filtering Technique used**	**No.of Particles**	**RMSE**
1	Extended Kalman filter	N/A	0.0368
2	Extended Kalman filter with smoother	N/A	0.0249
3	Unscented Kalman filter	N/A	0.0370
4	Unscented Kalman filter with smoother	N/A	0.0252
5	Particle filter	1000	0.0714
6	Particle filter	10000	0.0387
7	Particle filter with smoother	1000	0.0690
8	Particle filter with smoother	10000	0.0335

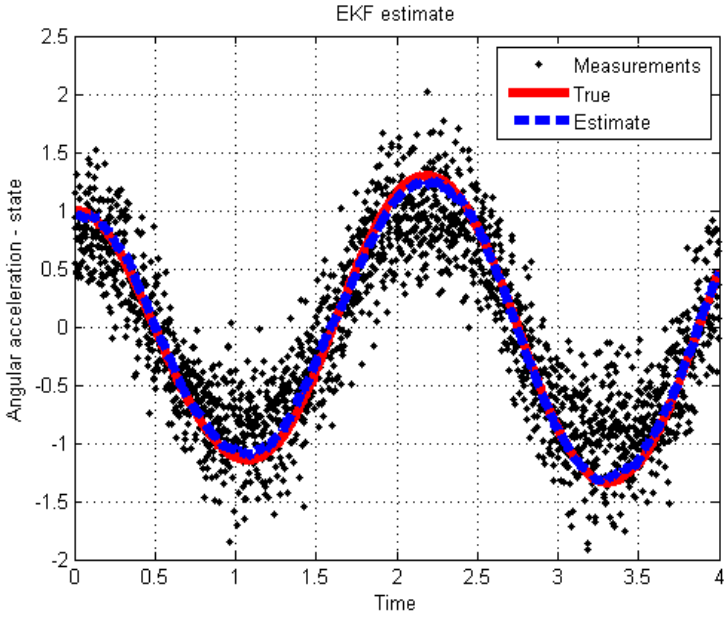

Fig. 18. State of noisy pendulum with extended Kalman filtering

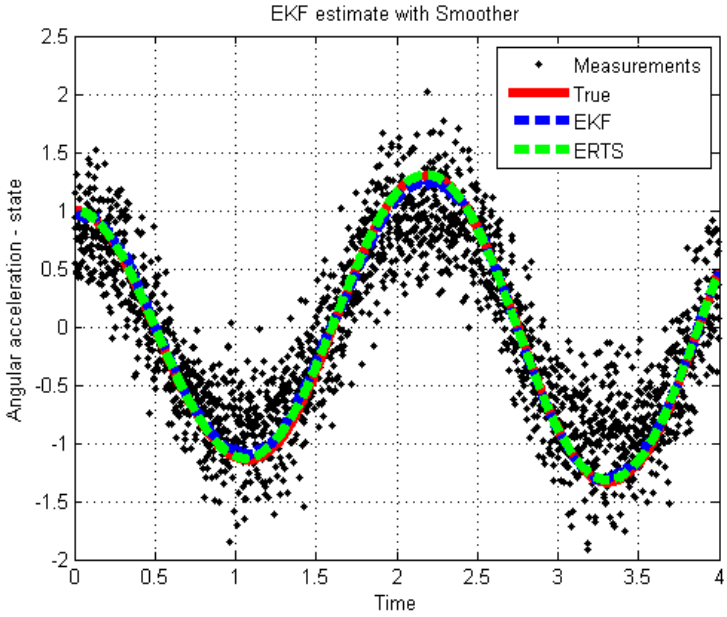

Fig. 19. State of noisy pendulum with extended Kalman filtering along with a smoothing distribution

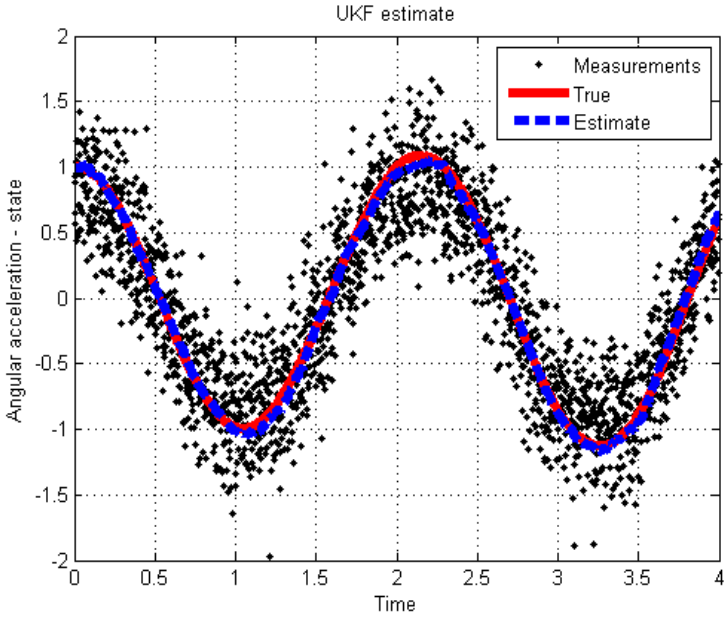

Fig. 20. State of noisy pendulum with unscented Kalman filtering

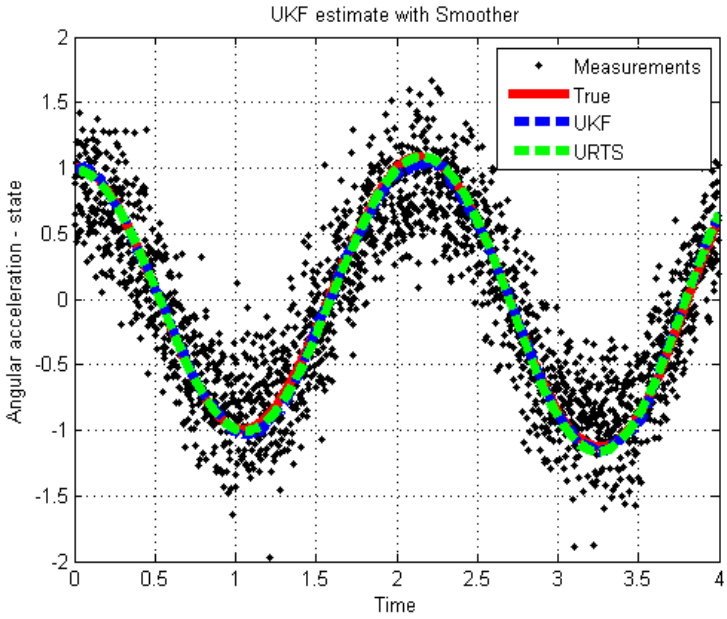

Fig. 21. State of noisy pendulum with unscented Kalman filtering along with a smoothing distribution

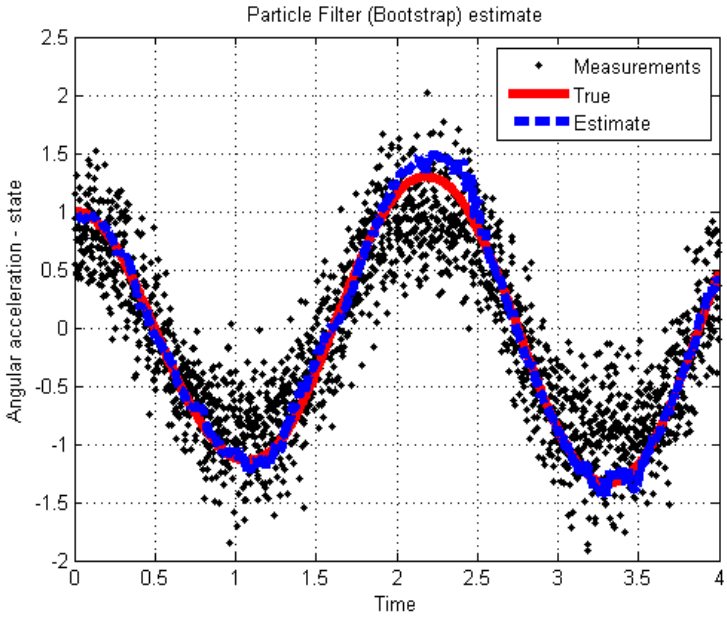

Fig. 22. State of noisy pendulum with particle filtering (no. of particles = 1000)

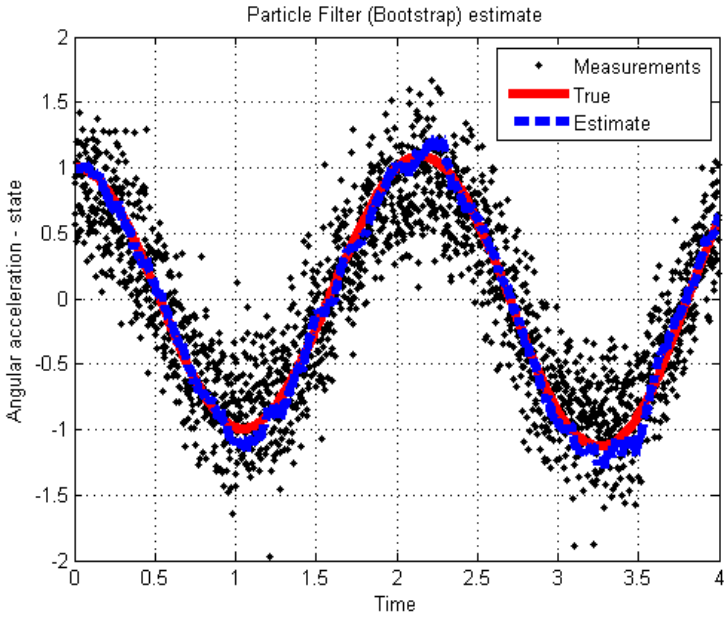

Fig. 23. State of noisy pendulum with particle filtering (no. of particles = 10000)

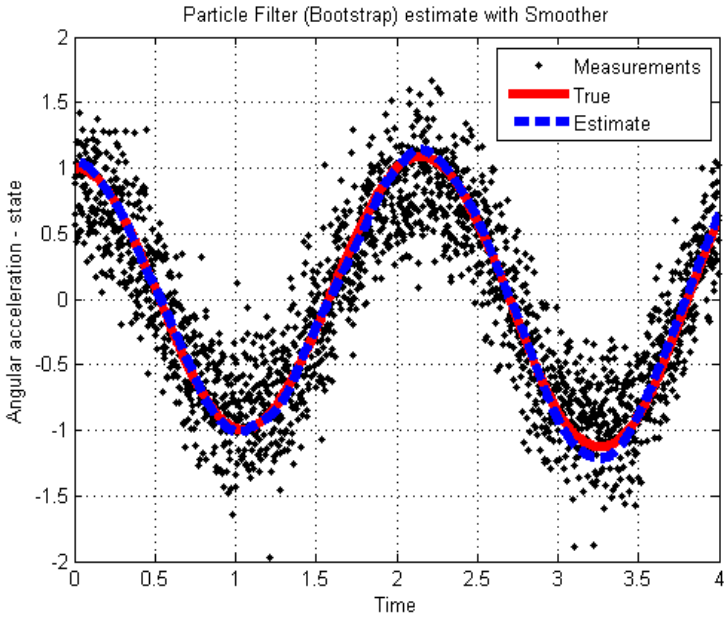

Fig. 24. State of noisy pendulum using particle filtering along with smoothing distribution (no. of particles = 1000)

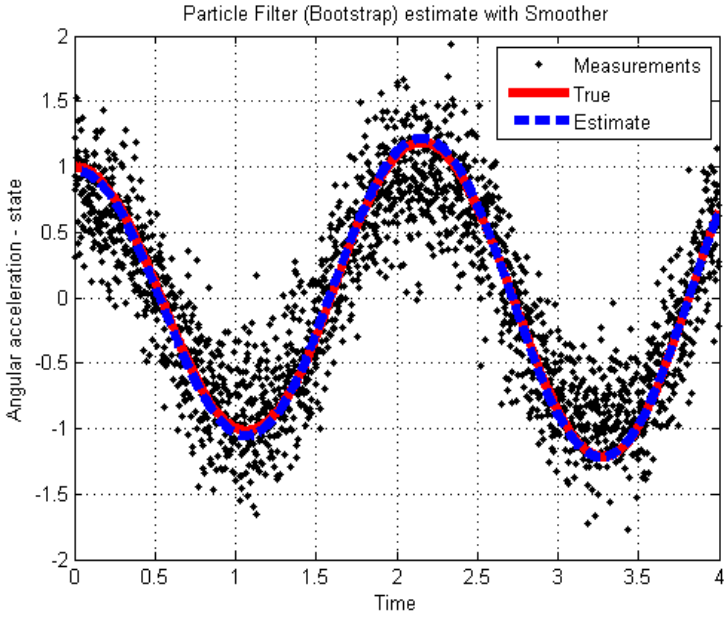

Fig. 25. State of noisy pendulum using particle filtering along with smoothing distribution (no. of particles = 10000)

CHAPTER VII

CALCULATION OF CAPACITY OF OPTICAL CHANNEL

The work of Shannon [27] on the information carrying capacity was extended to the optical fiber channel by Gordon et al. [40] who showed that amplified spontaneous emission (ASE) can be represented by AWGN fields. The classical theorem of Shannon [27] states that the capacity of a power-constrained transmission in an AWGN channel grows logarithmically with the increase of the signal to noise ratio (SNR). But, the non-linear fiber channel has a continuous injection of Gaussian noise based on ASE and the interaction between the signal and the noise results is very complicated non-Gaussian noise [41]. Recent progress in fiber optics has attracted fresh interest to the information theory of non-Gaussian non-linear communication channels.

The application of Shannons theory to the optical channel faces many challenges. A non-linear channel, unlike a linear channel, can create new frequencies which fall outside the spectral range of the input signal. The main non-linearity in the optical fiber channel is due to the Kerr effect [16]. Since this non-linearity depends on the signal power, one finds that increasing the signal power does not necessarily increase the information rate as is the case with a linear channel. Other impairments in the fiber channel include noise generation by ASE and chromatic dispersion caused by wavelength dependent propagation speed.

In this chapter, the calculation of capacity limits of a fiber channel in optically routed networks is discussed. First the calculation of a single optical channel is considered for QAM and Ring constellations. Then it is extended to a multi-channel optical WDM network. This work is done based on the work of Essiambre et al. [16]. All elementary Kerr nonlinear interactions in the presence of signal and noise are taken into account. The transmission through the fiber is described by direct numerical

solution of the stochastic generalized nonlinear Schrodinger equation (GNSE) using Split Step Fourier Transform method.

A. Fiber Propagation

Fiber optic communications typically operates in a regime of weak non-linearity and this allows one to use some standard techniques to address to the Shannon capacity of fiber optic communication [16]. Such a regime limits spectral broadening, or the process of creation of new frequencies. This regime of operation is often referred to as pseudolinear transmission [42].

There are two important sources of noise resulting from transmission over optical fibers: double Rayleigh back-scattering (DRB) [43] and amplified spontaneous emission (ASE). The back-scattered light for DRB propagates over a significant fiber length in the backward direction before being scattered back in the forward direction. This requires the insertion of optical isolators along the line to suppress the backward propagation of DRB. It can be shown from the expressions of single Rayleigh back-scattering [43] that for distributed amplification with gain compensating fiber loss, the ratio of DRB power to signal power is $\propto 1/N$, where N is the number of optical isolators in the line. Therefore, for a large number of isolators, the DRB can be shown to be not a fundamental source of noise as compared to the ASE.

Now, consider the evolution of the optical signal $E(z,t)$ through the fiber using distributed Raman amplification with ASE generation. This process can be shown mathematically using the GNSE,

$$\frac{\partial E}{\partial z} + \frac{i}{2}\beta_2 \frac{\partial^2 E}{\partial t^2} - i\gamma|E|^2 E = in(z,t) \tag{7.1}$$

where, β_2 represents the chromatic dispersion factor. The parameter γ is the instan-

taneous Kerr nonlinearity coefficient [42]. Reverse propagation is used by setting the right-hand side of Eq (7.1) to zero and changing z to $-z$. All signal-signal nonlinear interactions occurring within one WDM channel are undone by this process. The term $n(z,t)$ in Eq (7.1) is the term describing ASE noise generation. In [40], Gordon et al. showed ASE using the statistical properties of additive Gaussian noise.

$$\sigma_n^2 = n_{sp} K_T h \nu_s \alpha \tag{7.2}$$

where n_{sp} is the spontaneous emission factor, ν_s is the optical frequency of the signal, and α is the fiber loss coefficient. The parameter $K_T = 1 + \eta(T)$ is the phonon occupancy factor. This is close to 1 for Raman amplification of fiber-optic communication systems. Let P_s is the signal power of the WDM channel of interest. Now, for signals using a single state of polarization, the SNR and OSNR are simply related by $SNR = (2B_{ref}/S)OSNR$.

B. Input Alphabet and Modulation Scheme

In this work, two types of constellations are used. First the M-QAM constellation is used where M is the number of symbols being used. Later a multi-level symbol constellation is used which will be in the form of a concentric N-ring structure with equal amplitude spacing and random phase spacing (PSK). Each ring will have equal population of symbols. The modulation considered in this study uses Nyquist signals having box-like spectra with a square-root raised cosine shape and a roll-off of 20% [16]. The optical multiplexer and demultiplexer transfer functions are identical and match the square-root raised cosine signal spectrum. The magnitude of a modulated input pulse is seen in Fig. 26. The raised cosine roll-off is chosen to reduce the large memory in the time domain associated with perfectly square spectrum modulation

using the sinc temporal function [16].

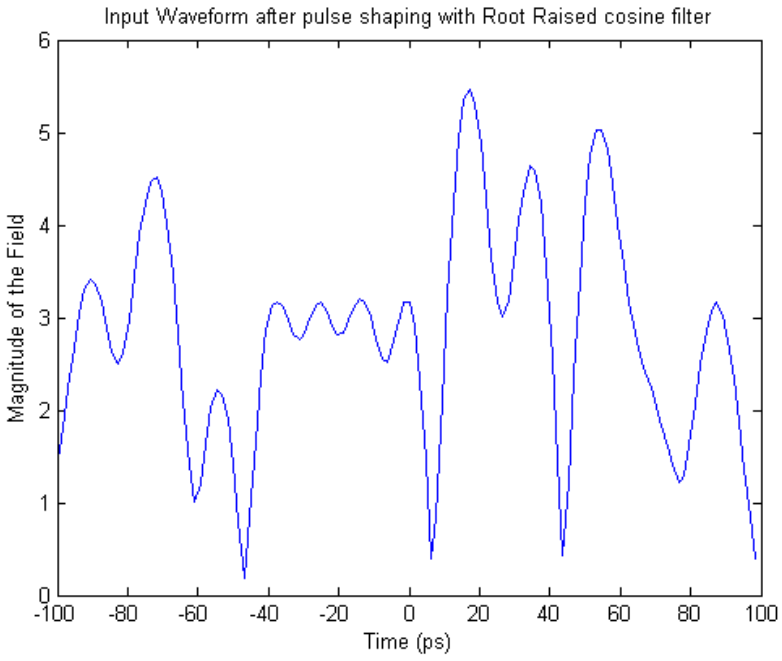

Fig. 26. Input field waveform with root raised cosine pulse-shaping

C. Capacity Calculation

The channel capacity, that we discussed in the Chapter V, for a specified channel input alphabet, when X is a random input giving rise to the random channel output Y, is given by

$$R = I(X;Y) = H(Y) - H(Y|X) \tag{7.3}$$

where the units are taken to be bits/sec/Hz. The functions $H(Y)$ and $H(Y|X)$ are referred to as the entropy of Y and the entropy of Y conditioned on X, respectively. For the numerical evaluation of fiber capacity, the channel is treated as a discrete memoryless channel (DMC) and using such a model removes the memory associated with signal-signal intra-channel nonlinearities. Nevertheless, ignoring the channel memory

and using a DMC model results in a lower bound on the achievable information rate. The discretized version of Eq (7.3) is applied for the case of both QAM input alphabet and the concentric ring input alphabet by summing up the entropies on the RHS of Eq (7.3).

However, for calculating the empirical entropies of Eq (7.3), the transmission of optical symbols through the optical fiber should be simulated which is not straightforward because of the presence of a non-linear partial differential equation. For this purpose, several numerical methods exists. These methods can be classified into two broad categories known as, (i) the finite-difference methods; and (ii) the pseudospectral methods. The one method that has been used extensively for this purpose is the split-step Fourier method [44]. The efficiency of this method is attributed in part to the use of the FFT algorithm. This method will be discussed in the subsequent section. The later sections will show the simulation results of the work.

1. Split Step Fourier Transform Method

To understand this method better, let the Eq (7.3) be written as in [45],

$$\frac{\partial E}{\partial z} = -\frac{i}{2}\beta_2 \frac{\partial^2 E}{\partial t^2} - \frac{\alpha}{2} + i\gamma|E|^2 E = [\hat{D} + \hat{N}]E \tag{7.4}$$

where $\hat{D}$ is a differential operator that accounts for dispersion in a linear medium given by $\hat{D} = -(i/2)\beta_2 \partial^2/\partial t^2 - \alpha/2$ and $\hat{N}$ is a non-linear operator that accounts for fiber non-linearities given by $\hat{N} = i\gamma|E|^2$. Both the linear and the non-linear parts may be solved independently, but the equation with both the parts may not be solved as easily. So an approximate sub-optimal solution needs to be used to solve such an equation.

The SSFM obtains the sub-optimal solution by treating the propagation of the optical field over a small distance h, thus enabling to assume both linear and non-

linear parts to act independently. Thus, the propagation from z to $z+h$ can be carried out independently in two steps with one step assuming $\hat{D}=0$ and the next step assuming $\hat{N}=0$. Mathematically this can be represented as,

$$E(z+h,t) \approx \exp(h\hat{D})\exp(h\hat{N})E(z,t) \tag{7.5}$$

The differential exponent term can be evaluated analytically using the fourier transform solution. This may be shown mathematically as,

$$\exp(h\hat{D})E(z,t) = F_T^{-1}\exp(h\hat{D}(i\omega))\tilde{E}(z,\omega) \tag{7.6}$$

where F_T denotes the Fourier-transform operation and $\hat{D}(i\omega)$ is the fourier transform pair obtained by changing the differential operator $\frac{\partial}{\partial t}$ by $i\omega$ and ω is the frequency in the Fourier domain. Also, $\tilde{E}(z,\omega)$ is the fourier transform of $E(z,t)$. This accounts for the propagation of the optical pulse through a small step h. By repeating the above N times, the pulse can be propagated over a length of Nh. The value of N is chosen so that the pulse propagates through the entire fiber. Moreover, it can be shown that the splitstep Fourier method is accurate to second order in the step size h.

The accuracy of the split-step Fourier method can be improved by using a different procedure of propagating the pulse from z to $z+h$. In this method the Eq (7.5) is varied as following,

$$E(z+h,t) \approx \exp(\frac{h}{2}\hat{D})\exp(\int_z^{z+h}\hat{N}(z`)dz`)\exp(\frac{h}{2}\hat{D})E(z,t) \tag{7.7}$$

The main difference here is that the effect of nonlinearity is included in the middle of the segment. Because of this symmetric form of the exponential operators, this scheme is known as the symmetrized split-step Fourier method [44]. The most important advantage of this new method is that the it is accurate to third order in

the step size h. As a result, this will provide more accurate solution to the problem. However, the calculation of the integral is not simple as can be seen from the following equation,

$$\int_{z}^{z+h} \hat{N}(z`)dz` = \frac{h}{2}[\hat{N}(z) + \hat{N}(z+h)] \tag{7.8}$$

From Eq (7.8), it is clear that the value of $\hat{N}(z+h)$ is unknown at the midsegment located at $z + h/2$ and so an iterative procedure is required. Although the iteration is time-consuming, it can still reduce the overall computing time based on the value of h.

Specifically, for the optical pulse propagation, the fiber length is divided into a large number of segments. The optical field $E(z,t)$ is first propagated for a distance $h/2$ with dispersion only using the FFT algorithm. At the midplane $z + h/2$, the nonlinear term that represents the effect of nonlinearity over the whole segment length h is then multiplied. Finally, the field is propagated the remaining distance of $h/2$ with dispersion only to obtain $E(z+h,t)$.

Though this method has its advantages, it requires that step sizes in z and T be selected carefully to maintain the accuracy. The optimum choice of step sizes depends on the complexity of the problem. In general, the split-step Fourier method is a very efficient method provided it is used properly.

D. Simulations

Before the actual capacity calculations are done, the propagation of optical wave needs to be simulated. That is, the transmission of information across the optical fiber needs to be simulated. For this purpose, consider a Gaussian pulse to be transmitted across the fiber. The Gaussian pulse forms a very simple example for understanding the

transmission. The Gaussian pulses are of the form of

$$U(0,T) = \exp(-\frac{T^2}{2T_0^2}), \tag{7.9}$$

where T_0 is the half-width (at 1/e-intensity point). In practice, the full width at half maximum (FWHM) is generally used for T_0. For a Gaussian pulse, the two are related by

$$T_{FWHM} = 2(ln2)^{1/2}T_0 = 1.665T_0. \tag{7.10}$$

Using these equations and substituting in the GNSE we can be able to simulate the transmission of such a pulse. For the simulation of GNSE, the method of Split Step Fourier Transform, discussed in previous sections, was used. Fig. 27 represents the

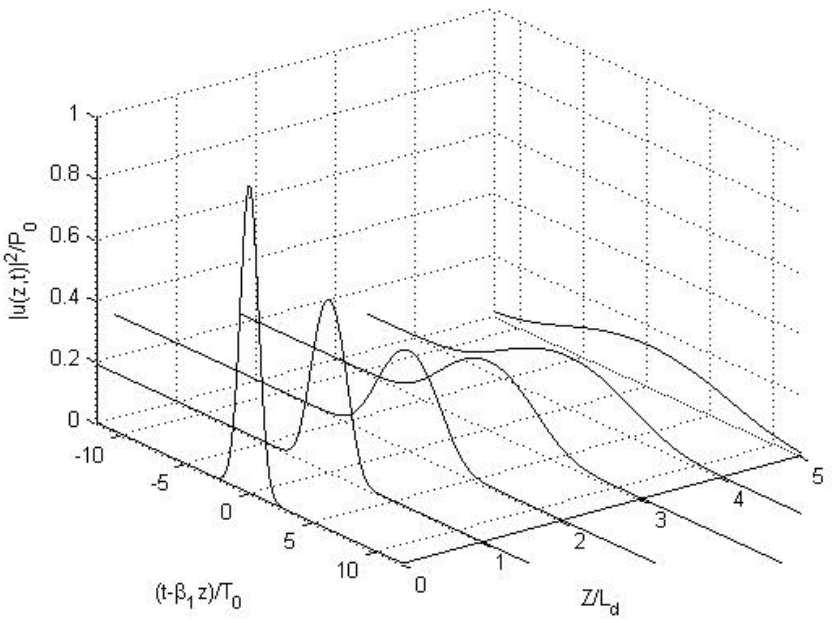

Fig. 27. Propagation of Gaussian pulse across fiber

transmission of such a Gaussian pulse through the fiber. For simplicity the fiber is shown in five sections. It shows the variations in the travelling wave as it progresses

across the sections of the fiber. The pulse broadening effect as seen in the Gaussian pulse is mainly due to the group velocity dispersion (GVD). The effect of GVD is that it changes the phase of each spectral component of the pulse. The amount of the phase change depends on both the frequency and the propagated distance. Even though such phase changes do not alter the pulse spectrum, they can distort the pulse shape [44]. Fig. 28 represents the reverse transmission of the Gaussian pulse through

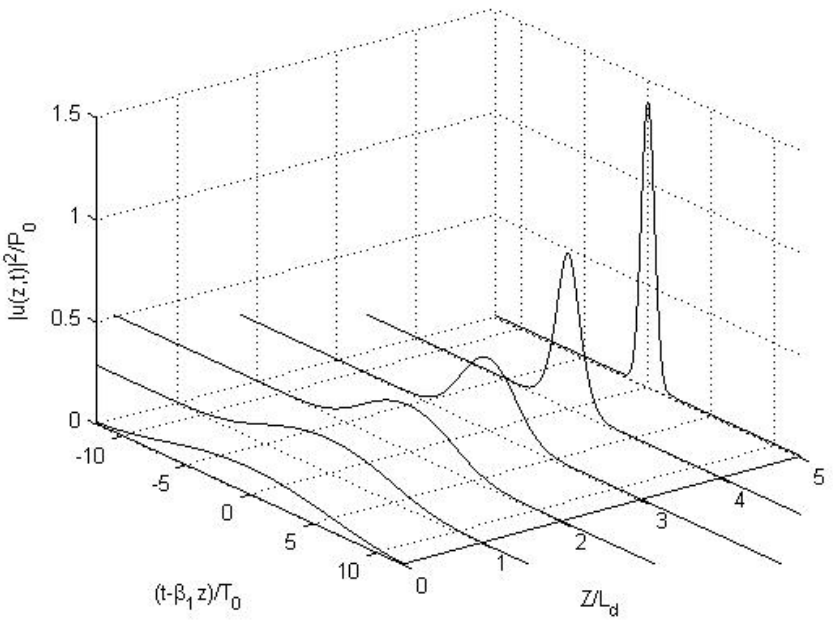

Fig. 28. Reverse propagation of Gaussian pulse across fiber

fiber. It is obtained directly by replacing dz by -dz in the split step fourier method. Since there is no noise assumed in the transmission the reverse transmission of the distorted wave will produce the actual input waveform that is transmitted [44]. This result is used later to find the capacity of the optical channel in the presence of ASE noise. The assumption that is taken for this purpose is that the value of ASE noise is very minimal (of the order of 10^{-10}) that the reverse propagation provides a good estimate for the transmitted signal. Fig. 29 shows the input wave along with the

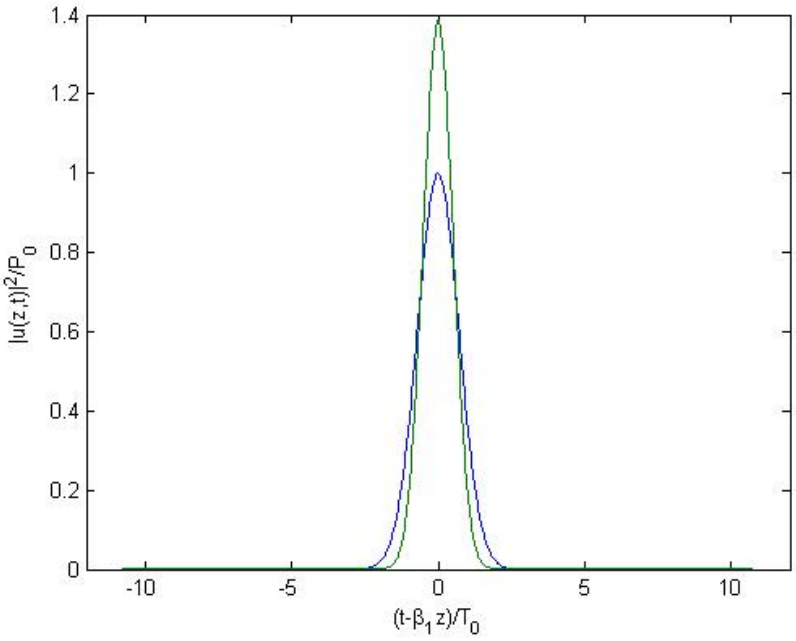

Fig. 29. Comparison of input and reverse propagated waveforms

pulse obtained by reverse propagation. It is clear that the reverse propagated wave provides a perfect estimate for the input wave when no noise is assumed. Also there is a decrease in the amplitude for the reverse propagated wave as compared to the input wave. As the fiber length increases the estimated wave diminishes to zero. For this purpose, several amplifiers are used at regular intervals so that this effect will be reduced. But these amplifiers also introduce the Amplifier stimulated emission (ASE). Now, using the method described in the previous sections, the capacity of the optical channel is calculated. Fig. 30 represents the estimated capacity for the channel with the input modulation scheme of 16-QAM. The root raised cosine pulse shaping filter is used along with a matched filter. It can be seen that as SNR increases the capacity increases with it as expected. Fig. 31 represents the estimated capacity for a WDM channel with five channels seperated in frequency. The input modulation scheme used is 16-QAM. The root raised cosine pulse shaping filter is used along

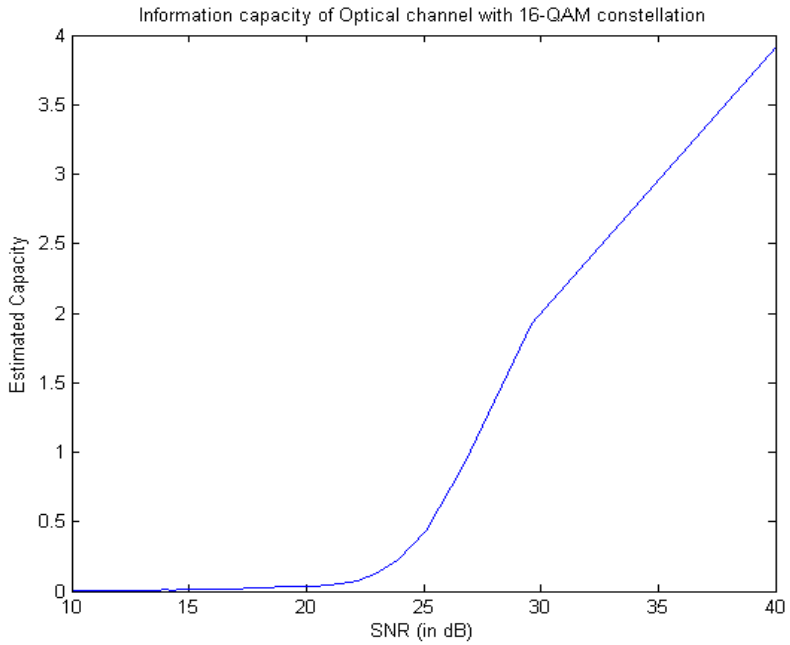

Fig. 30. Capacity of optical channel vs SNR with 16-QAM modulation scheme

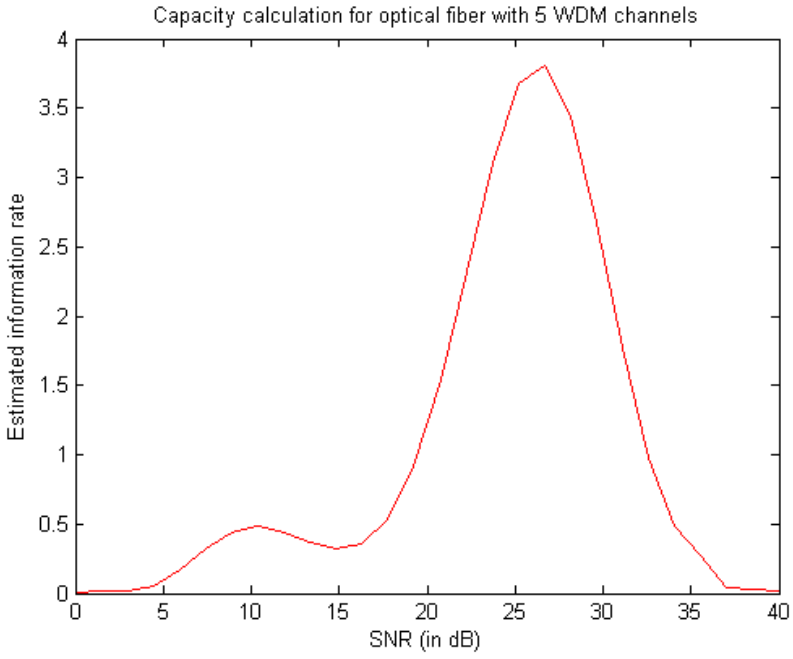

Fig. 31. Capacity of optical WDM channel vs SNR with 16-QAM modulation scheme and 5 channels

with a matched filter. It can be seen that as SNR increases the capacity increases till certain SNR is reached. From then, the effect of cross phase modulation (XPM) will become high and thus decreasing the capacity as SNR increases further. This effect of XPM depends on the number of WDM channels being used.

Thus, in this chapter, an algorithm for finding the capacity of optical channel is implemented and the effect of multiple WDM channels on the capacity is seen.

CHAPTER VIII

CONCLUSIONS

Typically, optimal estimation for non-linear non-Gaussian state-space models do not have an analytic solution. Since the development of particle filtering in 1993, these methods have become a very popular class of algorithms to solve the estimation problems numerically. The main advantage for these methods is that they can be carried out in an online manner, i.e., recursively as observations become available. Now-a-days, these methods are being routinely used in several fields such as communications, signal processing, computer vision, econometrics, robotics and navigation.

Many statistical signal processing problems found in wireless communications involves making inference about the transmitted data given the received signal in the presence of various unknown channel distortions. The optimal solution to these problems are often computationally complex to implement. This paved way to the use of several sub-optimal algorithms to tackle this problem. The particle filtering method is one such sub-optimal algorithm that can be used in this case. It uses the concept of sequential importance sampling (SIS) for the recursive computation of *a posteriori* distribution by drawing of samples from the importance density with corresponding importance weights. This work is aimed to introduce the applications of particle filtering to several communication channels.

First, the classical theories of discrete-time optimal filtering and the bayesian techniques have been reviewed. Then the particle filtering method is introduced and a generic algorithm is discussed as given in [2]. The example provided in [2] is also explained which helps in understanding the basic concepts better. The subsequent chapters discuss the applications of the particle filtering algorithm for communication channels.

We started off by discussing about the usage of particle filtering algorithm for calculating the bit error rate (BER) for transmission over an adaptive flat fading wireless channel as developed in [6]. We have used the state space model approach for deriving the particle filtering algorithm for the blind detection with the use of Kalman filtering algorithm. As simulation results show, the particle filtering method performs well when compared to the differential detection method for both Gaussian and non-Gaussian additive noise at high SNR. Typically, at BER of 10^{-2} there is a performance advantage of nearly 12 dB for a zero delay particle filter. For the delayed weighted scheme, an improvement in performance by about 5 dB at a BER of 10^{-3} is seen and it almost achieves the known channel bound.

Then we discussed the application of particle filtering algorithm in finding the mutual information rate of a particular communication channel as developed by Loeliger and Dauwels [14]. We extended this concept and applied it to the adaptive channel discussed above. We developed an algorithm and found the capacity of that channel using a particle filtering approach. The simulation results are shown at the end of the chapter.

Later, we discussed the particle filtering algorithms for non-linear and non-Gaussian continuous-discrete systems, that is, for the recursive Bayesian estimation of states of a stochastic differential equation observed through discrete-time measurements. These methods were initially developed by Simo Sarkka in his dissertation work [15]. The continuous-discrete unscented filter and smoother are introduced and they provide good alternatives to the extended Kalman filter and smoother in models, where the Jacobian and Hessian matrices of the drift terms are not available. The example of computing the angular acceleration of a noisy simple pendulum, as explained in [15,34], is discussed. The results for simulations are provided.

In the last section of the thesis study, we discussed the optical communication

channel so as to study the feasibility of a particle filtering algorithm to detect optical signals through the channel and also an algorithm to calculate the capacity of the optical channel. But since the noise present in an optical environment is of the magnitude of 10^{-24}, the need for such a particle filtering algorithm is dubious. There are more general algorithms, as discussed in this thesis work, which do the job of the optical detection. It is a good future work though to consider the particle filtering problem in this optical scenario in more detail.

The advantages of the particle filters make them an attractive alternative to the standard non-linear filters for efficient state and parameter estimation in all the applications with a similar problem statement.

REFERENCES

[1] Z. Chen, "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond," Adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada, 2003.

[2] S. Arulampalam, S. Maskell, N.J. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking" *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[3] B. Ristic, S. Arulampalam, and N.J. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Boston, MA: Artech House, 2004.

[4] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," in *IEE Proceedings F, Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.

[5] A. Papoulis, S.U. Pillai, *Probability, Random Variables and Stochastic Processes*, New York: McGraw-Hill, 4th edition, 2002.

[6] S. Haykin, *Adaptive Filter Theory*, Upper Saddle River, NJ: Prentice Hall, 4th edition, 2002.

[7] A. Doucet, S. Godstill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, pp. 197-208, 2000.

[8] A. Doucet and X. Wang, "Monte Carlo Methods for Signal Processing," *IEEE Signal Processing Magazine*, vol. 22, pp. 152-170, November 2005.

[9] S. Haykin, K. Huber, and Z. Chen, "Bayesian Sequential State Estimation for MIMO Wireless Communications," in *Proceedings of the IEEE*, vol. 92, no. 3, pp. 439-454, March 2004.

[10] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine*, vol. 20, pp. 19-38, September 2003.

[11] Y. Liu, B. Wang, W. He, J. Zhao, and Z. Ding, "Fundamental Principles and Applications of Particle Filters," in *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, 21-23 June, 2006, vol. 2, pp. 5327-5331.

[12] M. Simandl, O. Straka, "Sampling Densities of Particle Filter: A Survey and Comparison," in *Proceedings of the American Control Conference*, New York, 11-13 July, 2007, pp. 4437-4442.

[13] J.S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamical Systems," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1032-1044, 1998.

[14] J. Dauwels and H.A. Loeliger, "Computation of Information Rates by Particle Methods," in *Proc. 2004 IEEE International Symposium on Information Theory*, Chicago, June 27-July 2, 2004, pp. 178-190.

[15] S. Sarkka, "Recursive Bayesian Inference on Stochastic Differential Equations," Ph.D. dissertation, Helsinki University of Technology, Espoo, Finland, April 2006.

[16] R.J. Essiambre, G.J. Foschini, P.J. Winzer, G. Kramer and E.C. Burrows, "The Capacity of Fiber-Optic Communication Systems," in *Proceedings of Optical*

Fiber Communication Conference (OFC 2008), San Diego, CA, February 2008, pp. 1-3.

[17] J. Miguez, and P.M. Djuric, "Blind Equalization by Sequential Importance Sampling," in *Proceedings of IEEE, ISCAS*, Phoenix, AZ, May 2002, pp. 845-848.

[18] D. Guo and X. Wang, "Blind Detection in MIMO Systems via Sequential Monte Carlo," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 453-464, April 2003.

[19] Z. Yang and X. Wang, "A Sequential Monte Carlo Blind Receiver for OFDM Systems in Frequency-Selective Fading Channels," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 271-280, February 2002.

[20] G.L. Stuber, J.R. Barry, S.W. Mclaughlin, Y. Li, M.A. Ingram and T.G. Pratt, "Broadband MIMO-OFDM Wireless Communications," *Proceedings of IEEE*, vol. 92, no. 2, pp. 271-294, February 2004.

[21] J.G. Proakis, *Digital Communications*, New York: McGraw-Hill, 4th edition, 2003.

[22] T.S.Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, Second edition, 2006.

[23] R. Chen and J.S. Liu, "Mixture Kalman Filters," *Journal of the Royal Statistical Society - Series B*, vol. 62, pp. 493-508, 2000.

[24] J.K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 686-693, November 1991.

[25] R. Chen, X. Wang and J.S. Liu, "Adaptive Joint Detection and Decoding in Flat-Fading Channels via Mixture Kalman Filtering," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2079–2094, September 2000.

[26] D. Middleton, "Non-Gaussian Noise Models in Signal Processing for Telecommunications: New Methods and Results for Class A and Class B Noise Models," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1129-1149, May 1999.

[27] C.E. Shannon, "A Mathematical Theory of Communication: Part I," *The Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[28] B. Holter, "On the Capacity of the MIMO Channel: A Tutorial Introduction," in *Proc. of IEEE Norwegian Symposium on Signal Processing*, Trondheim, Norway, October 2001, pp. 167-172.

[29] D. Arnold and H.A. Loeliger, "On the Information Rate of Binary-Input Channels with Memory," in *Proc. 2001 IEEE Int. Conf. on Communications*, Helsinki, Finland, June 11-14, 2001, pp. 2692-2695.

[30] V. Sharma and S. K. Singh, "Entropy and Channel Capacity in the Regenerative Setup with Applications to Markov Channels," in *Proc. 2001 IEEE Int. Symp. Information Theory*, Washington, DC, June 24-29, 2001, pp. 283-290.

[31] H.D. Pfister, J.B. Soriaga, and P.H. Siegel, "On the Achievable Information Rates of Finite State ISI Channels," in *Proc. IEEE Global Communications Conf. 2001*, San Antonio, TX, Nov. 2001, pp. 2992-2996.

[32] D. Arnold, H.A. Loeliger, and P.O. Vontobel, "Computation of Information Rates from Finite-State Source/Channel Models," in *Proc. 40th Annual Aller-*

ton Conference on Communication, Control, and Computing, Allerton House, Monticello, IL, October 24, 2002, pp. 457-466.

[33] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Information Theory*, vol. 20, pp. 284-287, March 1974.

[34] S. Sarkka, "On Sequential Monte Carlo Sampling of Discretely Observed Stochastic Differential Equations," in *Proc. Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, Sept 2006, pp. 21-24.

[35] A.H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.

[36] I. Karatzas and S.E. Shreve, *Brownian Motion and Stochastic Calculus*, New York: Springer, 1991.

[37] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, New York: Springer, 6th edition, 2003.

[38] K. Gopinath, *Stochastic Filtering Theory*, New York: Springer-Verlag, 1980.

[39] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805-1819, 2005.

[40] J.P. Gordon, W.H. Louisell, and L.R. Walker, "Quantum Fluctuations and Noise in Parametric Processes II," *Phys. Rev Lett.*, vol. 129, pp. 481-485, 1963.

[41] J.P. Gordon, L.R. Walker, and W.H. Louisell, "Quantum Statistics of Masers and Attenuators," *Phys. Rev. Lett.*, vol. 130, pp. 806-812, 1963.

[42] K. S. Turitsyn, S. A. Derevyanko, I. V. Yurkevich, and S. K. Turitsyn, "Information Capacity of Optical Fiber Channels with Zero Average Dispersion," *Phys. Rev. Lett.*, vol. 91, no. 20, pp. 1-4, 2003.

[43] R.J. Essiambre, G. Raybon, and B. Mikkelsen, *Optical Fiber Telecommunications IV*, New York: Academic Press, 2002.

[44] J. Bromage, P. J. Winzer, and R.J. Essiambre, *Raman Amplifiers and Oscillators in Telecommunications*, New York:Springer Verlag, 2003.

[45] G. P. Agrawal, *Nonlinear Fiber Optics*, Boston: Academic Press, 4th edition, 2007.

APPENDIX A

KALMAN FILTERING ALGORITHM

This section shows the proof that the density $p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})$ is Gaussian for the channel used in Chapter IV.

Consider the state-space model for the adaptive wireless flat fading channel,

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{A.1}$$

$$y_t = s_t\mathbf{h}^H\mathbf{x}_t + \sigma v_t \tag{A.2}$$

where $\{u_t\}$, $\{v_t\}$ are white complex Gaussian noise sequences with unit variance and independent real and imaginary components. Now to show that in this case,

$$p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) \sim N_c(\mu_t^{(j)}, var_t^{(j)}) \tag{A.3}$$

where,

$$\mu_t^{(j)} = a_i h^H F \eta_{t-1} \tag{A.4}$$

$$var_t^{(j)} = h^H K_t^{(j)} h + \sigma^2 \tag{A.5}$$

$$\eta_t^{(j)} = h^H \mathbf{F}\mathbf{x}_{t-1}^{(j)} \tag{A.6}$$

Proof: Consider the Kalman filtering algorithm given in Chapter II. Also the proof here is shown for a single particle and hence the j index is removed. According to the definition of the Kalman filter, the innovation term of the Kalman filter for this model can be given by,

$$I_t = y_t - s_t\mathbf{h}^H F\hat{(x)}_{t-1} \tag{A.7}$$

Now, the correlation matrix of the innovation process is given by,

$$\begin{aligned} R_t &= s_t \mathbf{h}^H K_t \mathbf{h} s_t^* + \sigma^2 \\ &= |s_t|^2 \mathbf{h}^H K_t \mathbf{h} + \sigma^2 \\ R_t &= \mathbf{h}^H K_t \mathbf{h} + \sigma^2 \end{aligned} \tag{A.8}$$

and the Kalman gain is given by,

$$g_t = K_t \mathbf{h} s_t^* R_t^{-1} \tag{A.9}$$

Now to progress any further, we need to calculate the state error mean and covariance matrix for prediction, filtering and update stage for which the knowledge of the estimated state is important. But we know from the definition of such an estimated state from the Kalman algorithm. Using the Kalman algorithm, the estimated state vector can be written as,

$$\hat{x}_t = \mathbf{F}\hat{x}_{t-1} + g_t I_t \tag{A.10}$$

Substituting Eq (2.14) and Eq (A.7) in this equation, we get,

$$\hat{x}_t = \mathbf{F}\hat{x}_{t-1} + \frac{1}{R_t}(y_t - s_t \mathbf{h}^H \hat{F(x)}_{t-1}) s_t^* K_t \mathbf{h} \tag{A.11}$$

This equation of estimated state vector is seen in the particle filtering algorithm given in the Chapter IV.

From the Eq (A.11) the predicted error correlation matrix can be given by,

$$K_t = \mathbf{F}\Sigma_{t-1}\mathbf{F}^H + \mathbf{g}\mathbf{g}^H \tag{A.12}$$

and the filtered state mean and correlation matrix is given by,

$$\begin{aligned}
\eta_t &= F\eta_{t-1} + \frac{1}{R_t}(y_t - s_t h^H \mathbf{F}\eta_{t-1})K_t\mathbf{h} \qquad \text{(A.13)}\\
\Sigma_t &= K_t - g_t s_t \mathbf{h}^H K_t \\
&= K_t - K_t\mathbf{h}s_t^* R_t^{-1} s_t \mathbf{h}^H K_t \\
&= K_t - \frac{1}{R_t}|s_t|^2 K_t \mathbf{h}\mathbf{h}^H K_t \\
\Sigma_t &= K_t - \frac{1}{R_t} K_t \mathbf{h}\mathbf{h}^H K_t \qquad \text{(A.14)}
\end{aligned}$$

Thus the mean of the density $p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})$ can be obtained from the innovation term and the measurement model of the system as,

$$\begin{aligned}
\mu_t &= E\left[y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right] \\
&= s_t h^H \mathbf{F}\eta_{t-1}|_{s_t=a_i} \\
&= a_i h^H \mathbf{F}\eta_{t-1} \qquad \text{(A.15)}
\end{aligned}$$

which is the result to be proved. Similarly the covariance value is given by,

$$\begin{aligned}
var_t &= var\left[y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right] \\
&= \mathbf{h}^H K_t \mathbf{h} + \sigma^2 \\
&= R_t \qquad \text{(A.16)}
\end{aligned}$$

Hence, the density $p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1})$ takes the form of,

$$p(y_t|s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) \sim N_c(\mu_t^{(j)}, var_t^{(j)}) \qquad \text{(A.17)}$$

which is the same equation seen as in the particle filtering algorithm for the detection of signal through the flat fading channel dealt in Chapter IV.

VITA

Sirish Boddikurapati received his Bachelor of Engineering degree in electronics and communication engineering from the Andhra University in 2007. He started his master's program at Texas A&M University in August 2007 and will receive his Master of Science degree in December 2009.

Mr. Sirish Boddikurapati may be reached at the Department of Electrical Engineering, 214 Zachry Engineering Center, College Station, Texas 77843-3128 with attention to Henry Pfister or by email at sirishchandra4u@gmail.com .

The typist for this thesis was Sirish Boddikurapati.