# SEQUENTIAL MONTE CARLO METHODS FOR BLIND DETECTION IN WIRELESS COMMUNICATIONS

**A DISSERTATION**

*Submitted in partial fulfillment of the requirements for the award of the degree*

*of*

**MASTER OF TECHNOLOGY**

**in**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**(With Specialization in Communication Systems)**

By

**SRIKANTH ISANAKA**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**
**ROORKEE -247 667 (INDIA)**

**JUNE, 2009**

# CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled, **"SEQUENTIAL MONTE CARLO METHODS FOR BLIND DETECTION IN WIRELESS COMMUNICATIONS"** towards the partial fulfillment of the requirements for the award of the degree of **Master of Technology** with specialization in **Communication Systems**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from July 2008 to June 2009, under the guidance of **Dr.D.K.MEHRA, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 29-06-09

Place: Roorkee

**SRIKANTH ISANAKA**

# CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 29-06-09

Place: Roorkee

**Dr. D. K. MEHRA,**

Professor, E&C Department,

IIT Roorkee,

Roorkee – 247 667 (India).

# ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my guide, **Dr. D. K. MEHRA** for his able guidance, valuable suggestions and constant attention. It is his constant encouragement that inspired me throughout my dissertation work. I consider myself fortunate to have my dissertation done under him.

Thanks are due to the Lab staff of Signal Processing Lab, Department of Electronics and Communication Engineering, IIT Roorkee for providing necessary facilities.

I am greatly indebted to all my course mates, who have graciously applies themselves to the task of helping me with ample morale support and valuable suggestions. Most of all I would like to thank my family. Finally, I would like to extend my gratitude to all those persons who directly or indirectly contributed towards this work.

**SRIKANTH ISANAKA**

# ABSTRACT

Estimating the state of the system from noisy measurements is being increasingly used in many application areas which include signal processing, communications, statistics and econometrics. Filtering is a way to achieve this by incorporating noisy observations as they become available with prior knowledge of the system model. Due to the dramatic increase in the number of users and their demand for more advanced services, the need for fast and accurate filtering techniques in digital communications, capable of coping with challenging transmission conditions, is becoming more and more prevalent.

Bayesian methods form a rigorous general frame work for dynamic state estimation problems. The central idea to this recursive Bayesian estimation is to determine the probability density function of the state vector of the systems conditioned on the available measurements. However, the optimal exact solution to this Bayesian filtering problem is intractable since it requires high dimensional integration. Kalman filter provides an optimal solution in case of linear systems and Gaussian noise. For practical nonlinear filtering applications, extended Kalman filter, which is based on an assumption of Gaussian noise, yields approximate solutions.

Particle filtering algorithms, which are developed independently in various engineering fields, provides a numerical solution to the non-tractable recursive Bayesian estimation problem in case of non-linear and non-Gaussian systems.

In this dissertation work, we have used the state space model approach for deriving the particle filtering algorithm for blind detection in various systems namely SISO, MIMO, OFDM, with the use of Kalman filtering algorithm. Particle filters are sequential Monte Carlo methods that use a point mass representation of probability densities in order to propagate the required statistical properties for state estimation.

MIMO-OFDM systems can achieve higher data rates over broadband wireless channels. The blind detection in differentially encoded MIMO-OFDM systems using particle filtering algorithm is also been exploited. For simulation MATLAB is used and it is demonstrated through simulation results that the performance of particle filtering approach to blind detection is close to the optimal MLSE receiver.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AR | Auto Regressive |
| ARMA | Auto Regressive-Moving Average |
| ASIR | Auxiliary Sampling Importance Resampling |
| AWGN | Additive White Gaussian Noise |
| BER | Bit Error Rate |
| BPSK | Binary Phase Shift Keying |
| CK | Chapman-Kolmogorov |
| CP | Cyclic Prefix |
| CSI | Channel State Information |
| CSW | Cumulative Sum of Weights |
| DFT | Discrete Fourier Transform |
| EKF | Extended Kalman Filter |
| FFT | Fast Fourier Transform |
| IDFT | Inverse Discrete Fourier Transform |
| ISI | Intersymbol Interference |
| MAP | Maximum a Posteriori |
| MC | Monte Carlo |
| MCMC | Markov Chain Monte Carlo |
| MIMO | Multiple-Input Multiple-Output |
| MLSE | Maximum Likelihood Sequence Estimation |
| MMSE | Minimum Mean Square Error |
| MPSK | M-ary Phase Shift Keying |
| OFDM | Orthogonal Frequency Division Multiplexing |

| | |
|---|---|
| PF | Particle Filter |
| QPSK | Quadrature Phase Shift Keying |
| QRD | QR Decomposition |
| RMSE | Root Mean Square Error |
| RPF | Regularized Particle Filter |
| SIR | Sampling Importance Resampling |
| SIS | Sequential Importance Sampling |
| SISO | Single-Input Single-Output |
| SMC | Sequential Monte Carlo |
| SNR | Signal to Noise Ratio |
| TDL | Time Delay Line |
| WER | Word Error Rate |
| ZF | Zero Forcing |

# Chapter 1
# INTRODUCTION

In many application areas which include signal processing, statistics, communications, and econometrics, it is required to estimate the state of the system from a noisy measurements made on the system [1,2]. Bayesian methods form a rigorous general frame work for dynamic state estimation. In Bayesian frame work all the unknown quantities are treated as random variables and a priori knowledge of the system being modeled is often available for the formulation of Bayesian models. The Bayesian approach is to construct a posterior probability density of the state based on all available information. Using Baye's theorem a posterior density can be computed from the prior distributions and the likelihood function [3]. 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 [4], 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 [2,5]. In nonlinear or non-Gaussian problems the closed form solution to the recursive Bayesian filtering problem is intractable since it requires high dimensional integration. Therefore, approximate nonlinear filters [1,4] have been proposed, which can be categorized into five types: (1) analytical approximations, (2) direct numerical approximations, (3) sampling based approaches, (4) Gaussian mixture filters, and (5) simulation based filters. .

In nonlinear case, the most common approach is extended Kalman filter (EKF)[2,5], which approximate the model by linearized version of it using Taylor series expansion and then use the optimal Kalman filter with this approximate model. This filter works well for weekly nonlinear system. For systems with high degree of non linearity further terms in Taylor series should be considered, which results in additional computational complexity [2,5]. The EKF assumes the Gaussian nature which is not always satisfied with the real systems. Real systems commonly include non-linear and non-Gaussian elements as well as high dimensionality. There are many practical applications with non-linear and non-Gaussian features namely, localization of robots, estimating noisy digital communications signals, image processing, and aircraft tracking using radar measurements [1,4]. Numerical integration [1,4] is another approach that could be used in non-linear, non-Gaussian cases but it is computationally too expensive to be used in practical applications.

Although the idea of Monte Carlo simulation [4] originated in the late 1940s, its popularity in the field of filtering started in 1993[6]. Roughly speaking, Monte Carlo technique [6,7,8] 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 the sequential Monte Carlo 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]. Sequential Monte Carlo methods found limited use in the past, except for the last decade, primarily due to their very high computational complexity and the lack of adequate computing resources of the time. The fast advances of computers in the recent years and outstanding potential of particle filters have made them a very active area of research. In particular, the sequential Monte Carlo approach has been used in parameter estimation and state estimation. This SMC approach is known variously as particle filtering [2,4,7,8,11,12,13], boot strap filtering, the condensation algorithm, interacting particle approximations and survival of the fittest.

Particle filtering is an emerging and powerful methodology particularly useful in dealing with non linear and non-Gaussian problems based on the concept of sequential importance sampling and Bayesian theory [2,4]. In comparison with standard approximation methods, such as the popular Extended Kalman Filter, the advantage of particle filtering is in that the exploited approximation does not involve linearizations around current estimates but rather approximates the representation of the desired distributions by discrete random measures [7,8]. Particle filters are sequential Monte Carlo methods which can be applied to any state space model and which generalizes the Kaman filtering methods. 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 [1,2,4,10]. Particle filtering methods are founded upon Monte Carlo simulations of the underlying systems and provide a convenient and attractive approach to computing the posterior distribution. Particle filtering methods are not limited by nonlinearity and non-Gaussianity constraints and can be implemented in a relatively simple fashion for a wide variety of problems. Due to the need for more complete physical models, the next generation of filtering methods will have to deal with nonlinear and non-Gaussian model components. Particle filtering methods have the potential to use the increasing computational power available in today's technological market to push filtering theory beyond its challenges.

## 1.1 Applications of Particle Filtering in Wireless Communications

Much of the world's communication technology today involves the transmission of digital signals over noisy channels, and thus also requires reliable real-time estimates of the system state. In communication systems, the maximization of the symbol a posteriori probability results in an optimal receiver in the sense that it minimizes the error probability. However, in situations of unknown quantities (such as channel and noise parameters), it usually becomes very difficult to construct the marginal a posteriori distribution in fading channels. Multipath fading results from the fact that radio signal propagates through many paths with different delays from transmitter to the receiver [14,15]. A novel adaptive Bayesian receiver for signal detection and decoding in fading channels with unknown channel statistics is presented in [16,17].

There is an increasing demand for the design of multiple-input multiple-output (MIMO) communication system for high data-rate wireless communications. An MIMO system employs multiple antennas at the transmitter and the receiver, and its capacity increases linearly with the minimum between the numbers of transmit and receive antennas [9,18]. In case of known channels, various detectors can be employed to demodulate the transmitted data symbols, such as the maximum likelihood sequence detector (MLSE)[14,18], the zero forcing detector[14,18], the minimum mean square error detector[14,18]. Receiver has to know the channel statistics for efficient demodulation of the received symbols. However, the channel dynamics cannot be known in advance and they will change from time to time. Various methods have been developed to know the unknown statistics of the channel. If the system is linear and the noise is approximated as Gaussian, then the Kalman filter will give the optimal solution. But if the power amplifiers feeding the transmit antennas are nonlinear then the state-space model of MIMO channel will be nonlinear [9,18]. Moreover, due to the non-Gaussian nature of the dynamic noise and measurement noise, the state space model of a MIMO channel is indeed non-Gaussian. A novel sequential Monte Carlo blind receiver for MIMO systems is presented in [19].

Orthogonal frequency division multiplexing (OFDM) [20] is one of the most promising techniques for achieving high speed wireless data communications. OFDM is a multicarrier transmission technique which divides the single wideband channel into a number of narrowband channels called sub-channels; each subcarrier in each sub-channel is being modulated by a low rate data stream and subcarriers are transmitted parallel over the channel [20,21]. The increased symbol duration reduces the impact of ISI. The main attraction of OFDM is based on its implementation using cost efficient Fast Fourier Transform (FFT) to implement multiple carrier modulation or demodulation operations. Thus the robustness to frequency-selective fading channels accompanied by high spectral efficiency and the feasibility of low cost transceiver implementations have led OFDM to being considered as a promising candidate for high data rate wireless communications [21]. A sequential Monte Carlo blind receiver for OFDM systems in frequency-selective fading channels is presented in [22].

MIMO system is used for high data rate transmission in a dense multi-path scattering environment, which causes the MIMO channel to be frequency-selective. OFDM can transform such a frequency-selective MIMO channel into a set of parallel frequency-flat MIMO channels, and reduces the ISI caused by the multipath. Thus MIMO-OFDM [23] systems are used as effective means of providing high-speed data transmission over dispersive wireless channels. The extension of particle filter approach for blind detection in MIMO-OFDM systems is considered in this dissertation work.

Recent advances (in) digital wireless communication technologies have allowed development of wireless sensor networks. Their use may span a vast range of fields, and their effectiveness is already being felt both in commercial and military applications as well as in the further development of science and engineering. Target tracking by particle filtering in binary sensor networks is presented in [24]. Particle filtering for positioning, navigation and tracking have been in presented in [25]. Particle filters finds applications in a wide variety of fields.

## 1.2 Statement of the Problem

To develop a blind Bayesian receiver for MIMO-OFDM communication systems over unknown fading channels. For this purpose, we explore the feasibility of using particle filtering approach to the blind detection in the above system by formulating it as a state space model i.e., state and observation equations. The receiver is based on the SMC methods for computing the a posteriori probabilities of unknown transmitted symbols. This Particle filtering technique does not require any training or pilot symbols and it can applied even the channel noise may be either Gaussian or Non-Gaussian. This dissertation presents the following work:

1. Study of Particle filters and its application to blind detection in SISO flat fading channel.
2. Application of Particle filtering in blind detection of MIMO systems in flat fading channel.
3. Application of Particle filtering in blind detection in OFDM systems over frequency selective channel.

4. Extensions of Particle filter approach to the MIMO OFDM systems.

## 1.3 Organization of the Report

This report is organized in six chapters:

In *chapter 1,* the overview of particle filters and its applications to wireless communications is presented and the statement of problem of the dissertation work is summarized.

In *chapter 2,* an overview of recursive Bayesian approach to the estimation of the system state using noisy measurements made on the system is described first. The optimal filtering technique namely Kalman filter, for the linear system and Gaussian noise is summarized. 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.

In *chapter 3,* the particle filtering approach for the blind detection in SISO and MIMO systems in flat fading channels is described. The simulation results are also presented for both SISO and MIMO systems.

In *chapter 4*, the particle filtering approach for the blind detection in OFDM systems in frequency-selective fading channels is presented. Simulation results are also given.

In *chapter 5,* the particle filter approach is extended to blind detection in MIMO-OFDM systems and simulation results are presented.

*Chapter 6* concludes the report with suggestions for future work.

# Chapter 2
# SEQUENTIAL MONTE CARLO METHODS FOR BAYESIAN FILTERING

In this chapter, an overview of recursive Bayesian approach to the estimation of the system state using noisy measurements made on the system is described first. The optimal filtering technique namely Kalman filter, for the linear system and 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. The degeneracy phenomenon in particle filter and the concept of resampling in particle filter is described next .The choice of sampling density with emphasis on the Gaussian optimal importance function is discussed. Another version of particle filter i.e., sampling importance resampling (SIR) filter is also presented. Finally the simulation results of SIR filter for a nonlinear system is presented.

## 2.1 Recursive Bayesian Estimation

Bayesian theory is a branch of probability theory that helps to model the uncertainty about the world and the outcomes of interest by incorporating prior knowledge and observational evidence. Bayesian analysis, interpreting the probability as a conditional measure, is one of the popular methods in many cases.

In Bayesian reference, all uncertainties (including states, parameters, which are either time-varying or fixed but unknown, priors) are treated as random variables. The inference is performed with in the Bayesian framework given all of available information. The objective of Bayesian inference is to use the priors and causal knowledge, quantitatively and qualitatively, to infer the conditional probability, given finite observations. There are usually three levels of the probabilistic reasoning in Bayesian analysis. Starting with model selection given the data and assumed priors; estimate the parameters to fit the data given the model and priors; and update the hyper parameters of the prior. There are three types of intractable problems inherently related to the evaluation of *a posteriori* density $p(\mathbf{x}/\mathbf{y})$ [1].

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

$$p(\mathbf{x}/\mathbf{y}) = \frac{p(\mathbf{y}/\mathbf{x})p(\mathbf{x})}{\int_X p(\mathbf{y}/\mathbf{x})p(\mathbf{x})d\mathbf{x}} \tag{2.1}$$

- **Marginalization:** Given the posterior $p(\mathbf{x},\mathbf{z}/\mathbf{y})$, the marginal posterior $p(\mathbf{x}/\mathbf{y})$ is calculated by

$$p(\mathbf{x}/\mathbf{y}) = \int_Z p(\mathbf{x},\mathbf{z}/\mathbf{y})d\mathbf{z} \tag{2.2}$$

- **Expectation:** Given the conditional pdf $p(\mathbf{x}/\mathbf{y})$, the expectation of the function $f(\mathbf{x})$ can be calculated as

$$E_{p(\mathbf{x}/\mathbf{y})}[f(\mathbf{x})] = \int_X f(\mathbf{x})p(\mathbf{x}/\mathbf{y})d\mathbf{x} \tag{2.3}$$

where **x**, **y** and **z** are random variables in equations (2.1), (2.2) and (2.3).

For many problems in communications and signal processing, an estimate is required every time a measurement is received. In this case, 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. State space model [2,5], which is used in such situations is essentially a notational convenience used for estimation and control problems. State space model comprises of two models namely system model and measurement model. The system model describes about the evolution of the state with time and measurement model relates the noisy measurements to the state. The generalized form of state space model is given by [2]

**System equation**

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{2.4}$$

where $\mathbf{f}_k : \Re^{n_x} \times \Re^{n_v} \to \Re^{n_x}$ is a possibly nonlinear evolution function.

$n_x$ and $n_v$ are the dimensions of the state and process noise respectively.

$\mathbf{x}_k \in \Re^{n_x}$ is state vector.

$\mathbf{v}_{k-1} \in \Re^{n_v}$ is an i.i.d process noise.

**Measurement equation:**

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{2.5}$$

where $\mathbf{h}_k : \Re^{n_x} \times \Re^{n_n} \to \Re^{n_z}$ is a possibly nonlinear measurement function

$n_x$ and $n_n$ are the dimensions of the state and measurement noise respectively

$\mathbf{n}_k \in \Re^{n_n}$ is an i.i.d measurement noise

From the Bayesian perspective of dynamic state estimation, it is required to construct a posterior probability density function (pdf) of the state $p(\mathbf{x}_k / \mathbf{z}_{1:k})$ based on all the available observations $\mathbf{z}_{1:k}$ up to time *k*. It is assumed that the initial pdf $p(\mathbf{x}_0 / \mathbf{z}_0) \equiv p(\mathbf{x}_0)$ of the state vector, which is also known as the prior, is available. Then the pdf $p(\mathbf{x}_k / \mathbf{z}_{1:k})$ may be obtained recursively in two stages: *prediction* and *update*. Two assumptions are used to derive the recursive Bayesian filter [1].

(i) The states follow a fi rst-order Markov process i.e.,

$$p(\mathbf{x}_k / \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k / \mathbf{x}_{k-1}) \tag{2.6}$$

(ii) The observations are independent of the given states.

### 2.1.1 Prediction Stage

The prediction stage uses the system model to predict the state pdf forward from one measurement time to next. Since the state is usually subject to unknown disturbances (modelled as random noise), prediction generally translates, deforms, and spreads the state pdf. Specifically, given the pdf $p(\mathbf{x}_{k-1}/\mathbf{z}_{1:k-1})$ which is already available at time $k$-1, this stage involves the calculation of the pdf $p(\mathbf{x}_k/\mathbf{z}_{1:k-1})$.

$$p(\mathbf{x}_k/\mathbf{z}_{1:k-1}) = p(\mathbf{x}_k,\mathbf{z}_{1:k-1})/p(\mathbf{z}_{1:k-1})$$

$$= \left(\int_{-\infty}^{\infty} p(\mathbf{x}_k,\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}\right)/p(\mathbf{z}_{1:k-1})$$

$$= \left(\int_{-\infty}^{\infty} p(\mathbf{x}_k/\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})p(\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}\right)/p(\mathbf{z}_{1:k-1})$$

$$= \left(\int_{-\infty}^{\infty} p(\mathbf{x}_k/\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})p(\mathbf{x}_{k-1}/\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}\right)$$

$$= \left(\int_{-\infty}^{\infty} p(\mathbf{x}_k/\mathbf{x}_{k-1})p(\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}\right) \qquad (\because \text{ assumption (i)})$$

$$p(\mathbf{x}_k/\mathbf{z}_{1:k-1}) = \left(\int_{-\infty}^{\infty} p(\mathbf{x}_k/\mathbf{x}_{k-1})p(\mathbf{x}_{k-1},\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}\right) \qquad (2.7)$$

The equation (2.7) is known as the Chapman-Kolmogorov (CK) equation [3].

### 2.1.2 Update Stage

The update stage involves modification of the prediction pdf based on the latest measurement available at that time. Specifically, given the measurement $p(\mathbf{z}_k)$ available at time $k$ then it is used to update the prior via Baye's rule [3].

$$p(\mathbf{x}_k/\mathbf{z}_{1:k}) = p(\mathbf{x}_k,\mathbf{z}_{1:k})/p(\mathbf{z}_{1:k})$$

$$=\left[p(\mathbf{z}_k/\mathbf{x}_k,\mathbf{z}_{1:k-1})p(\mathbf{x}_k,\mathbf{z}_{1:k-1})\right]/\left[p(\mathbf{z}_k/\mathbf{z}_{1:k-1})p(\mathbf{z}_{1:k-1})\right]$$

$$=p(\mathbf{z}_k/\mathbf{x}_k)p(\mathbf{x}_k/\mathbf{z}_{1:k-1})/p(\mathbf{z}_k/\mathbf{z}_{1:k-1})$$

$$p(\mathbf{x}_k/\mathbf{z}_{1:k})=p(\mathbf{z}_k/\mathbf{x}_k)p(\mathbf{x}_k/\mathbf{z}_{1:k-1})/p(\mathbf{z}_k/\mathbf{z}_{1:k-1}) \tag{2.8}$$

where the normalizing constant is given by

$$p(\mathbf{z}_k/\mathbf{z}_{1:k-1})=\left(\int_{-\infty}^{\infty}p(\mathbf{z}_k/\mathbf{x}_k)p(\mathbf{x}_k/\mathbf{z}_{1:k-1})d\mathbf{x}_k\right) \tag{2.9}$$

The normalizing constant depends on the likelihood function $p(\mathbf{z}_k/\mathbf{x}_k)$ defined by the measurement model and the known statistics of observation noise $\mathbf{n}_k$. In the update stage, the measurement $\mathbf{z}_k$ is used to modify the prior density to obtain the required posterior density of the current state. The predictor and update relations are shown in Fig.2.1.

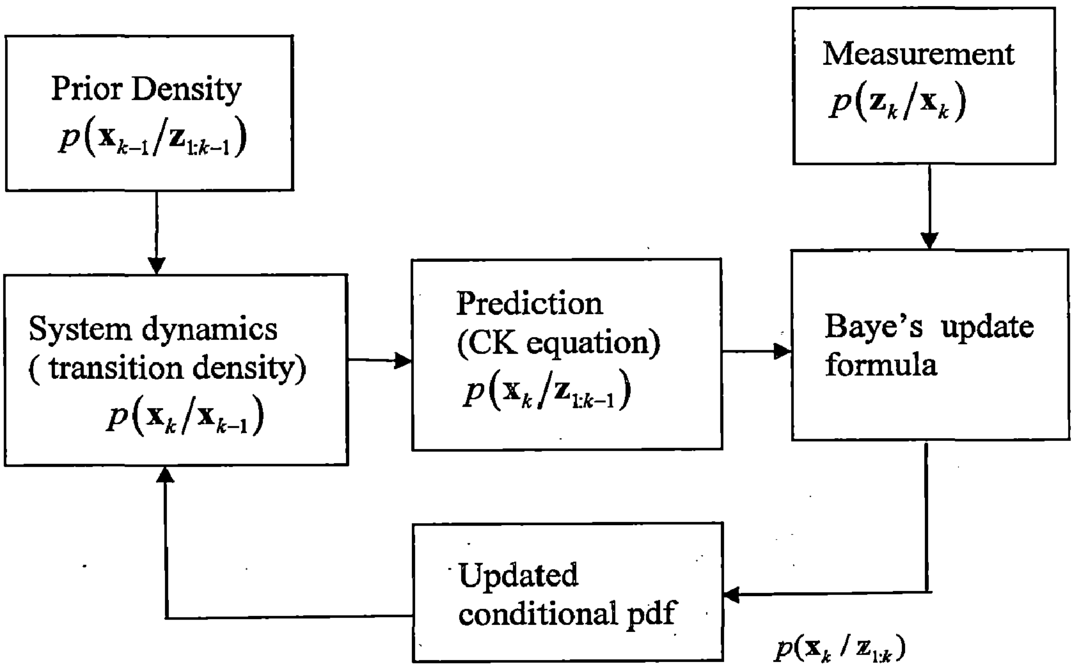



**Figure 2.1**. Prediction and update stages for the recursive Bayesian estimation

The recursive relations described above are easily solved for linear/Gaussian systems. In case the system is nonlinear/non-Gaussian in nature, the integrals are not tractable. In such cases, the approximate solution is provided by several non-linear filters.

## 2.2 Kalman Filter

Kalman filter [5] is a linear, discrete-time filter which can be applied to the stationary and nonstationary environments without any modification and its solution can be computed recursively. In particular, each updated estimate of the state is computed from the previous estimate and the new input data, so only previous estimate requires storage. In addition to eliminating the need for storing the entire past observed data, the Kalman filter is computationally more efficient than computing the estimate directly from all of those past data at each step of the filtering process. Kalman filter assumes that the posterior density at every time step is Gaussian and, hence parameterized by mean and covariance. To apply Kalman filter certain assumptions must hold. They are:

- $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are drawn from Gaussian distributions of known parameters.
- $\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$ is known and is a linear function of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$
- $\mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$ is a known linear function of $\mathbf{x}_k$ and $\mathbf{n}_k$

Thus the state equation (2.4) and measurement equation (2.5) are written as

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{2.10}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k \tag{2.11}$$

where $\mathbf{F}_k$ and $\mathbf{H}_k$ are known matrices defining linear functions. The covariances of $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$. Assume that the $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ have zero mean and are statistically independent. The Kalman filtering algorithm is summarized as follows [2]:

$$p(\mathbf{x}_{k-1} / \mathbf{z}_{1:k-1}) = N(\mathbf{x}_{k-1}; \mathbf{m}_{k-1/k-1}, \mathbf{P}_{k-1/k-1}) \tag{2.12}$$

$$p(\mathbf{x}_k / \mathbf{z}_{1:k-1}) = N(\mathbf{x}_k; \mathbf{m}_{k/k-1}, \mathbf{P}_{k/k-1}) \tag{2.13}$$

$$p(\mathbf{x}_k / \mathbf{z}_{1:k}) = N(\mathbf{x}_k; \mathbf{m}_{k/k}, \mathbf{P}_{k/k}) \tag{2.14}$$

where $N(\mathbf{x};\mathbf{m},\mathbf{P})$ is a Gaussian density with argument **x**, mean **m**, and covariance **P**.

$$\mathbf{m}_{k/k-1} = \mathbf{F}_k \mathbf{m}_{k-1/k-1} \tag{2.15}$$

$$\mathbf{P}_{k/k-1} = \mathbf{Q}_{k-1} + \mathbf{F}_k \mathbf{P}_{k-1/k-1} \mathbf{F}_k^T \tag{2.16}$$

$$\mathbf{m}_{k/k} = \mathbf{m}_{k/k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k/k-1}) \tag{2.17}$$

$$\mathbf{P}_{k/k} = \mathbf{P}_{k/k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k/k-1} \tag{2.18}$$

In addition, $\mathbf{S}_k$ the covariance of the innovation term $\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k/k-1}$, and $\mathbf{K}_k$ the Kalman gain respectively are given by

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{2.19}$$

$$\mathbf{K}_k = \mathbf{P}_{k/k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.20}$$

The Kalman filter algorithm 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 Kalman filter update is represented in Fig. 2.2.

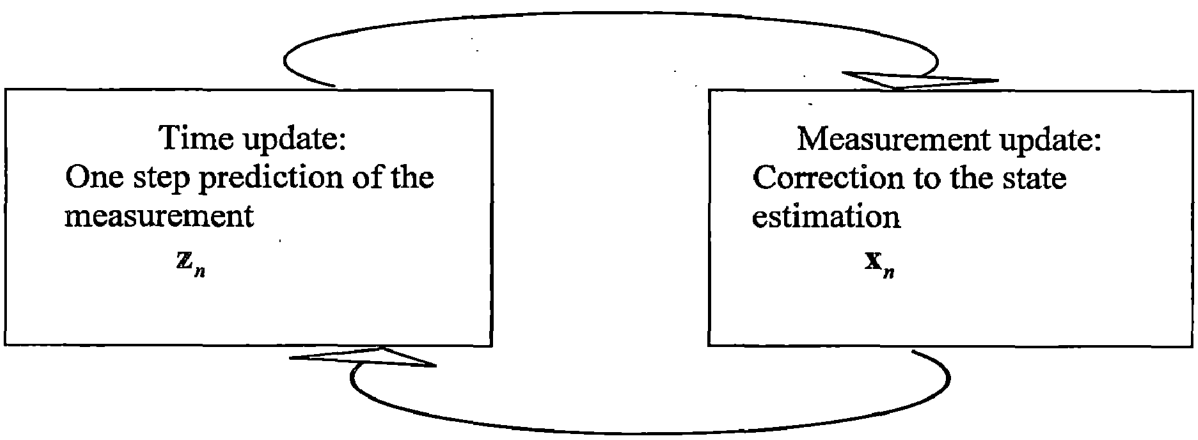



**Figure 2.2** Predictor-Corrector form of Kalman filter

## 2.3 Monte Carlo Sampling

Monte Carlo [MC] methods [1] are commonly used for approximation of intractable integrals and rely on the ability to draw a random sample from the required probability distribution Monte Carlo methods use statistical sampling and estimation techniques to evaluate the solutions to mathematical problems. Monte Carlo techniques have attracted lot of attention and have been developed in many areas. Monte Carlo methods have three categories: (i) Monte Carlo sampling, which is devoted to developing efficient sampling technique for estimation; (ii) Monte Carlo calculation, which is aimed  to design various random or pseudo-random number generators; and (iii) Monte Carlo optimization, which is devoted to applying the Monte Carlo idea to optimize some non differentiable functions. In the following only Monte Carlo sampling [4] is discussed.

Consider the multidimensional integral $I = \int g(\mathbf{x})d\mathbf{x}$, where $\mathbf{x} \in R^{n_x}$ .Monte Carlo methods for numerical integration factorize $g(\mathbf{x}) = \pi(\mathbf{x}) f(\mathbf{x})$ in such a way that $\pi(\mathbf{x})$ is interpreted as a probability density satisfying $\pi(\mathbf{x}) \geq 0$ and $\int \pi(\mathbf{x})d\mathbf{x} = 1$, $f(\mathbf{x})$ is an integrable function in a measurable space. The assumption is that it is possible to draw $N$ samples $\{\mathbf{x}^i; i = 1,....N\}$ distributed according to $\pi(\mathbf{x})$. Then the pdf $\pi(\mathbf{x})$ can be approximated as [4]

$$\pi(\mathbf{x}) = \frac{1}{N}\delta(\mathbf{x} - \mathbf{x}^i) \tag{2.21}$$

The Monte Carlo estimate of the integral

$$I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \tag{2.22}$$

is the sample mean

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \tag{2.23}$$

By taking large number of samples, the estimate converges to its true value. The variance of the estimate is inversely proportional to number of samples. There are several issues which are of concern in Monte Carlo sampling [1]

- **Consistency:** An estimator is consistent if the estimator converges to the true value almost surely as the number of observations approaches infinity.
- **Unbiasedness:** An estimator is unbiased if its expected value is equal to the true value.
- **Efficiency:** An estimator is efficient if it produces the smallest error covariance matrix among all unbiased estimators, it is also regarded optimally using the information in the measurements. A well-known efficiency criterion is the Cramer-Rao bound.
- **Robustness:** An estimator is robust if it is insensitive to the gross measurement errors and the uncertainties of the model.
- **Minimal variance:** Variance reduction is the central issue of various Monte Carlo approximation methods, most improvement techniques are variance reduction oriented.

In Bayesian estimation context, density $\pi(\mathbf{x})$ is the posterior density. It is not possible to sample effectively from the posterior distribution, being multivariate, nonstandard, and only known up to proportionality constant. A possible solution is to apply the importance sampling method.

### 2.3.1 Importance Sampling

Ideally the samples are generated from the density $\pi(\mathbf{x})$ and the integral $I$ is evaluated by using $I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)$. If the samples are easily generated from a density $q(\mathbf{x})$, which is similar to $\pi(\mathbf{x})$, then a correct weighting of the sample set still makes the Monte Carlo estimation possible. The pdf $q(\mathbf{x})$ is referred to as importance or proposal density [4,7,8,9]. Its similarity to $\pi(\mathbf{x})$ is interpreted by

$$\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0 \qquad \forall \mathbf{x} \in R^{n_x} \tag{2.24}$$

This means that $q(\mathbf{x})$ and $\pi(\mathbf{x})$ has same support. The equation (2.24) is necessary for the importance sampling theory to hold and, if valid, the integral $I$ is written as

$$I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \tag{2.25}$$

provided that $\frac{\pi(\mathbf{x})}{q(\mathbf{x})}$ is upper bounded. A Monte Carlo estimate of $I$ is computed by generating $N$>>1 independent samples $\{\mathbf{x}^i; i=1,....N\}$ distributed according to $q(\mathbf{x})$ and forming the weighted sum:

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)\hat{w}(\mathbf{x}^i) \tag{2.26}$$

where,

$$\hat{w}(\mathbf{x}^i) = \frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)} \qquad i=1,.......,N \tag{2.27}$$

are the *importance weights*[4]. If normalizing factor of the desired density $\pi(\mathbf{x})$ is unknown, then normalization of the importance weights is carried out. Then the estimate of the integral $I_N$ is given by

$$I_N = \frac{\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)\hat{w}(\mathbf{x}^i)}{\frac{1}{N}\sum_{j=1}^{N}\hat{w}(\mathbf{x}^j)} = \sum_{i=1}^{N} f(\mathbf{x}^i)w(\mathbf{x}^i) \tag{2.28}$$

where, the normalized importance weights are given by

$$w(\mathbf{x}^i) = \frac{\hat{w}(\mathbf{x}^i)}{\frac{1}{N}\sum_{j=1}^{N}\hat{w}(\mathbf{x}^j)} \qquad i=1,.......,N \tag{2.29}$$

This technique is used in the Bayesian framework, where $\pi(\mathbf{x})$ is the posterior density.

## 2.4 Particle Filtering

Sequential Monte Carlo methods have found limited use in the past, except for the last decade, primarily due to their very 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. Particle filter [4,6,7,8,9,10,11,12] 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. Particle filters are sequential Monte Carlo methods which can be applied to any state space model and which generalizes the Kaman filtering methods. The advantage of particle filtering over other methods is in that the exploited approximation does not involve linearizations around current estimates but rather approximations in the representation of the desired distributions by discrete random measures. Particle filter is best suited for nonlinear state-space models and non-Gaussian noises. Particle filters 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 etc.

### 2.4.1 Sequential Importance Sampling (SIS)

In order to make Bayesian importance sampling more practical, it will be convenient to calculate the particle weights recursively. The sequential importance sampling (SIS) [2,4,6,7,8,11] 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 $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ denote a random measure that characterizes the posterior pdf $p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k})$. Where, $\{\mathbf{x}_{0:k}^i, i = 0,....,N_s\}$ is a set of sample points with associated weights $\{w_k^i, i = 1,....,N_s\}$ and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0,....,k\}$ is the set of all states up to time $k$. The weights are normalized such that $\sum_i w_k^i = 1$. By SIS algorithm, the set $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ is recursively computed from the set $\{\mathbf{x}_{0:k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}$ when a new measurement $\mathbf{z}_k$ is available at time $k$. Specifically, suppose at time $k-1$ the posterior pdf $p(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1})$ is approximated by a random measure $\{\mathbf{x}_{0:k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}$, then SIS algorithm builds a random measure by appending newly generated particles $\mathbf{x}_k^i$ to the $\mathbf{x}_{0:k-1}^i$ and updating the weights $w_k^i$ to form $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ that properly represent the posterior pdf $p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k})$ Then, the posterior density at time $k$ is approximated as [2]

$$p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta\left(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{\mathbf{i}}\right) \tag{2.30}$$

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

$$w_k^i \propto \frac{p(\mathbf{x}^i{}_{0:k} / \mathbf{z}_{1:k})}{q(\mathbf{x}^i{}_{0:k} / \mathbf{z}_{1:k})} \tag{2.31}$$

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

$$q(\mathbf{x}_{0:k} / \mathbf{z}_{1:k}) = \frac{q(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})}{q(\mathbf{z}_{1:k})}$$

$$= \frac{q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})}{q(\mathbf{z}_{1:k})}$$

$$= \frac{q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{z}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) q(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})}{q(\mathbf{z}_k / \mathbf{z}_{1:k-1}) q(\mathbf{z}_{1:k-1})}$$

$$= \frac{q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{z}_k / \mathbf{z}_{1:k-1}) q(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})}{q(\mathbf{z}_k / \mathbf{z}_{1:k-1}) q(\mathbf{z}_{1:k-1})}$$

$$= q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1})$$

$$q(\mathbf{x}_{0:k} / \mathbf{z}_{1:k}) = q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1}) \tag{2.32}$$

By the equation (2.32), the samples $\mathbf{x}_{0:k}^i \sim q(\mathbf{x}_{0:k} / \mathbf{z}_{1:k})$ are obtained by augmenting each of the existing samples $\mathbf{x}_{0:k-1}^i \sim q(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}_k^i \sim q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. The pdf $p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k})$ is expressed as

$$p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})}{p(\mathbf{z}_{1:k})}$$

$$= \frac{p(\mathbf{z}_k / \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k / \mathbf{z}_{1:k-1}) p(\mathbf{z}_{1:k-1})}$$

$$= \frac{p(\mathbf{z}_k / \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k / \mathbf{z}_{1:k-1}) p(\mathbf{z}_{1:k-1})}$$

$$= \frac{p(\mathbf{z}_k / \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k / \mathbf{z}_{1:k-1})}$$

$$= \frac{p(\mathbf{z}_k / \mathbf{x}_k) p(\mathbf{x}_k / \mathbf{x}_{k-1}) p(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k / \mathbf{z}_{1:k-1})}$$

$$p(\mathbf{x}_{0:k} / \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k / \mathbf{x}_k) p(\mathbf{x}_k / \mathbf{x}_{k-1}) p(\mathbf{x}_{0:k-1} / \mathbf{z}_{1:k-1}) \tag{2.33}$$

where $p(\mathbf{z}_k / \mathbf{z}_{1:k-1})$ is a normalized constant. Now substituting the equations (2.32) and (2.33) in equation (2.31), then

$$w_k^i \propto \frac{p(\mathbf{z}_k / \mathbf{x}^i_k) p(\mathbf{x}^i_k / \mathbf{x}^i_{k-1}) p(\mathbf{x}^i_{0:k-1} / \mathbf{z}_{1:k-1})}{q(\mathbf{x}^i_k / \mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}^i_{0:k-1} / \mathbf{z}_{1:k-1})}$$



$$w_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k / \mathbf{x}^i_k) p(\mathbf{x}^i_k / \mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k / \mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k})} \qquad (2.34)$$

Furthermore, if $q(\mathbf{x}_k / \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k / \mathbf{x}_{k-1}, \mathbf{z}_k)$, then the importance density becomes only dependent on $\mathbf{x}_{k-1}$ and $\mathbf{z}_k$ This is particularly useful in the common case when only a filtered estimate of $p(\mathbf{x}_k / \mathbf{z}_{1:k})$ is required for each time step. In such situations, only $\mathbf{x}_k^i$ need to be stored and the path $\mathbf{x}_{0:k-1}^i$ , the history of observations $\mathbf{z}_{1:k-1}$ can be discarded. Then the modified weight is given by

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k / \mathbf{x}_k^i) p(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)} \qquad (2.35)$$

The posterior filtered density $p(\mathbf{x}_k / \mathbf{z}_{1:k})$ is given by

$$p(\mathbf{x}_k / \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right) \qquad (2.36)$$

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 by algorithm 2.1 [2, 4].

***Algorithm 2.1:*** *SIS Particle filter*

$$\left[\left\{\mathbf{x}_k^i, w_k^i\right\}_{i=1}^{N_s}\right] = SIS\left[\left\{\mathbf{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^{N_s}, \mathbf{z}_k\right]$$

- *FOR* $i = 1 : N_s$

  - *Draw* $\mathbf{x}_k^i \sim q(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$

➢ *Assign each particle with the importance weight up to a normalizing constant according to*

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k / \mathbf{x}_k^i) p(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$$

- *END FOR*
- *Calculate the total weight: t=SUM* $\left[\left\{\tilde{w}_k^i\right\}_{i=1}^{N_s}\right]$
- *FOR* $i = 1 : N_s$

   ➢ *Normalize the weights:* $w_k^i = t^{-1}\tilde{w}_k^i$

- *END FOR*

### 2.4.2 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 [2,4,11,12].The variance of the importance weights increases over time, and thus it is 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] ($N_{eff}$) given by

$$N_{eff} = \frac{N_s}{1 + Var(w_k^{*i})} \tag{2.37}$$

Where, $w_k^{*i} = p(\mathbf{x}_k^i / \mathbf{z}_{1:k}) / q(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$.Thus the effective sample size cannot be evaluated exactly, an estimate is calculated instead which is given by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2} \tag{2.38}$$

The small $N_{eff}$, the severe will be the degeneracy. There are three basic measures to mitigate the degeneracy problem in particle filters, (1) by increasing the number of samples $N_s$, (2) resampling. (3) by good choice of importance density. The simplest method to mitigate the degeneracy effect is to use a very large $N_s$. however it will increase the computational load on the system, which is often impractical.

## Resampling

Effects of degeneracy in particle filter is reduced by using resampling [13,26,27,28,29] where the particles having small weights are eliminated and the particles with large weights are replicated. The resampling stage is depicted in Fig.2.3 [10].

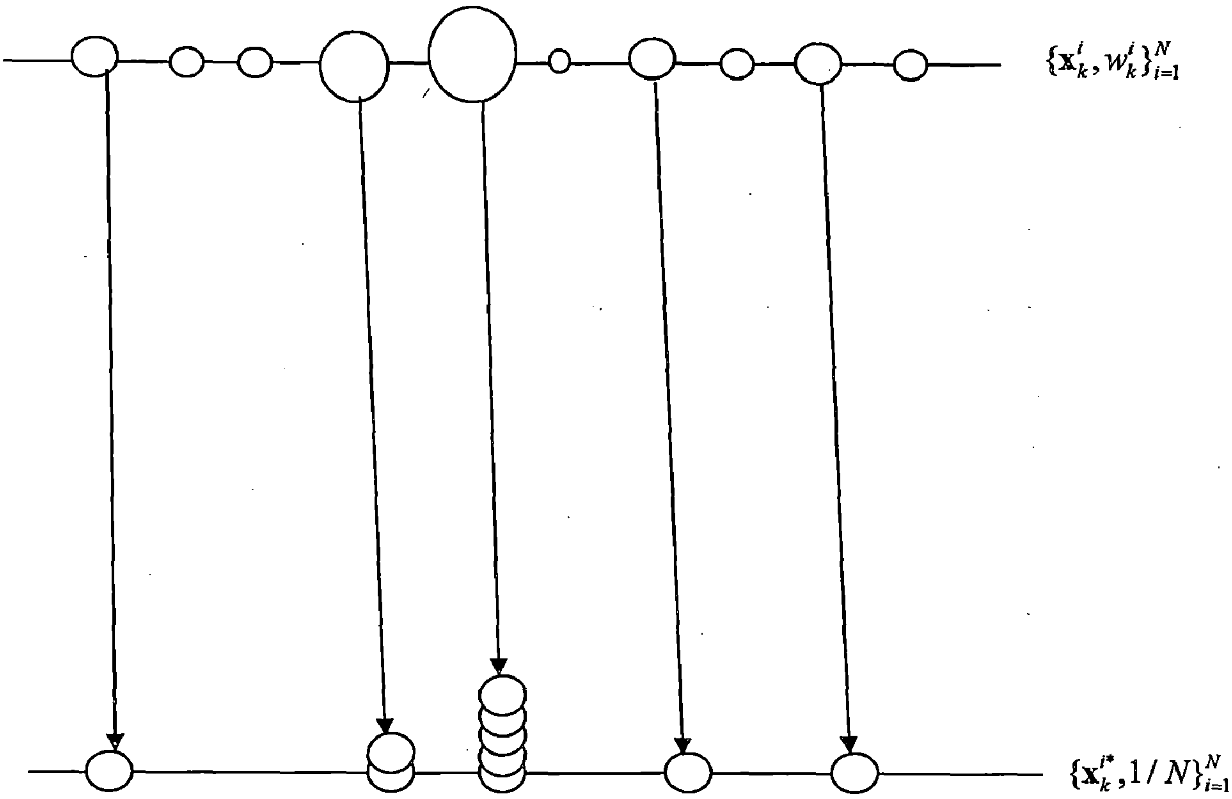



**Figure 2.3** Particle resampling

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. The resampling stage involves drawing of $N$ samples from the a posterior pdf with replacement. 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. From Fig.2.3, 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.

Resampling involves a mapping of random measure $\{\mathbf{x}_k^i, w_k^i\}$ into a random measure $\{\mathbf{x}_k^{i*}, 1/N\}$ with uniform weights. The set of random samples $\{\mathbf{x}_k^{i*}\}_{i=1}^{N}$ is generated by resampling (with replacement) $N$ times from an approximate discrete representation of $p(\mathbf{x}_k/\mathbf{z}_{1:k})$ with the probability $p\{\mathbf{x}_k^{i*} = \mathbf{x}_k^j\} = w_k^j$. The resulting sample is an i.i.d sample from the a posterior density $p(\mathbf{x}_k/\mathbf{z}_{1:k})$, and hence new weights are uniform.. The selection of new samples is schematically shown in the Fig.2.4 [4].

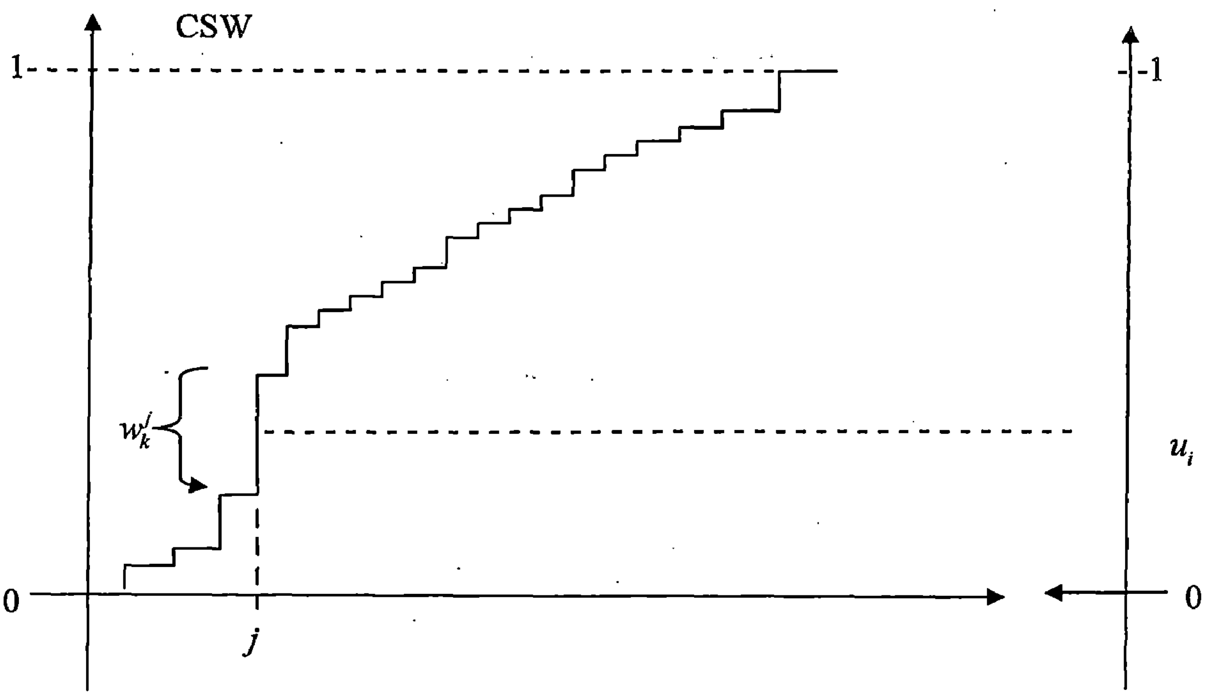



**Figure 2.4** The process of resampling

In Fig.2.4, the acronym CSW stands for the cumulative sum of weights of the random measure $\{\mathbf{x}_k^i, w_k^i\}$, and random variable $u_i, i=1,....,N$ is uniformly distributed in the interval [0,1]. From Fig.2.4, the main idea in the process of resampling is to select the new particles by comparing an ordered set of uniformly distributed random numbers $u_i, i=1,....,N$ lies in the interval [0, 1] with the cumulative sum of the normalized weights. It may be seen that from Fig.2.4, uniform random variable $u_i$ maps into index $j$ and the corresponding particle $\mathbf{x}_k^j$ has a good chance of being selected and multiplied because of its high value of $w_k^j$. This technique is mainly used in systematic resampling [4,27,28] and residual resampling [13,27,28], which are given by algorithm 2.2 and 2.3.

***Algorithm 2.2:*** *Systematic resampling* [27,28]

- *Generate N uniform random numbers*

$$\tilde{u} \sim U[0,1)$$

- *Obtain the N ordered random numbers* $u_k$

$$u_k = \frac{(k-1)+\tilde{u}}{N}$$

- *Allocate the* $n_i$ *copies of the particle* $x_i$ *to the new distribution*

$$n_i = \text{the number of } u_k \in \left[\sum_{s=1}^{i-1} w_s, \sum_{s=1}^{i} w_s\right)$$

***Algorithm 2.3:*** *Residual resampling* [27,28]

- *Allocate* $n_i' = \lfloor N w_i \rfloor$ *copies of the particle* $x_i$ *to the new distribution*
- *Additionally, resample* $m = N - \sum n_i'$ *particles from* $\{x_i\}$ *by making* $n_i''$ *copies of particle* $x_i$ *where the probability for selecting* $x_i$ *is proportional to* $w_i' = N w_i - n_i'$ *using systematic resampling.*

Now, the generic particle filter algorithm is given by algorithm 2.4.

***Algorithm 2.4:*** *Generic particle filter* [2, 4]

$$\left[\left\{\mathbf{x}_k^i, w_k^i\right\}_{i=1}^{N_s}\right] = PF\left[\left\{\mathbf{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^{N_s}, \mathbf{z}_k\right]$$

- *FOR* $i=1:N_s$
  - *Draw* $\mathbf{x}_k^i \sim q(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - *Assign each particle with the importance weight up to a normalizing constant according to*

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k / \mathbf{x}_k^i) p(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$$

- *END FOR*
- *Calculate the total weight: t=SUM*$\left[\left\{\tilde{w}_k^i\right\}_{i=1}^{N_s}\right]$
- *FOR* $i=1:N_s$
  - *Normalize the weights:* $w_k^i = t^{-1}\tilde{w}_k^i$
- *END FOR*
- *Calculate effective sample size* $\hat{N}_{eff}$ *by*

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2}$$

- *IF* $\hat{N}_{eff} < N_T$
  - *Resample using systematic resampling or residual resampling.*
- *END IF*

The general particle filtering algorithm 2.4 may be represented by Fig.2.5 [5].

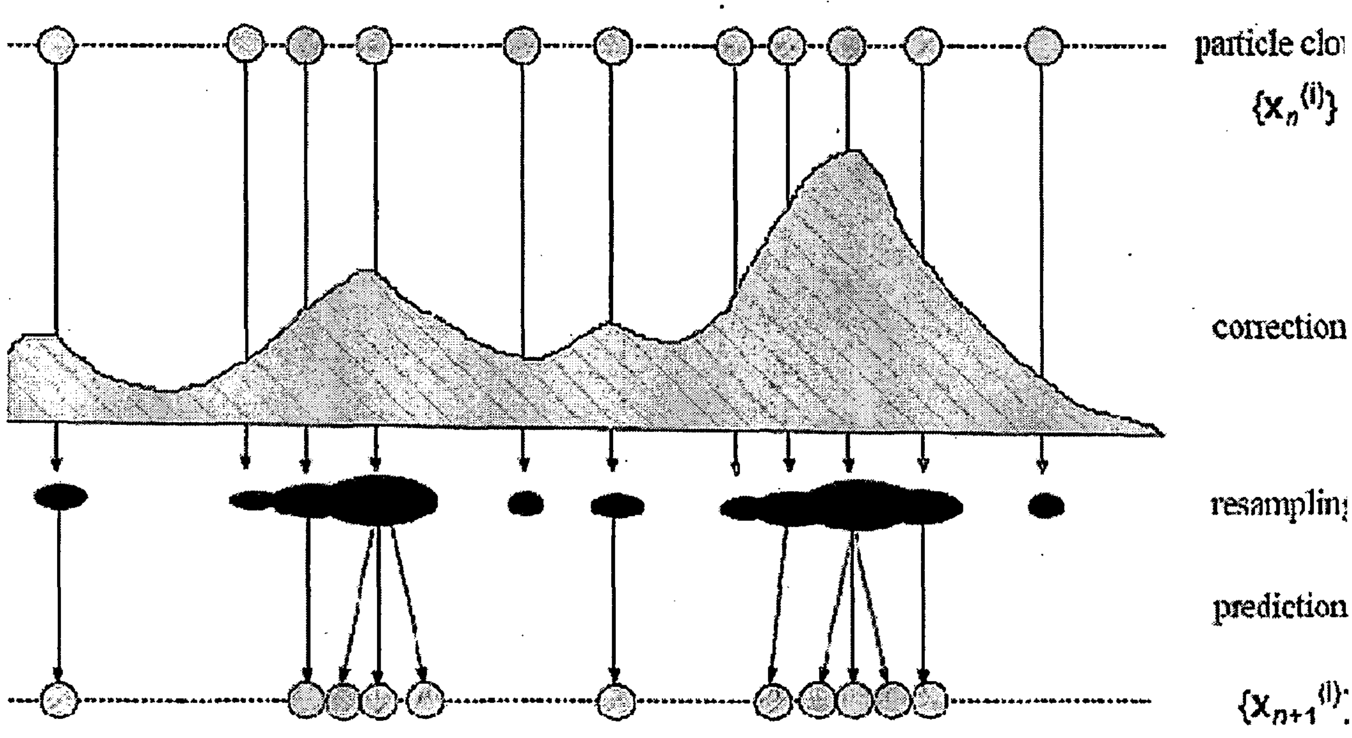



**Figure 2.5** An illustration of generic Particle filter with importance sampling and resampling

From Fig.2.5, it is seen that the particles are modified by the importance density function. The higher the probability, the denser the particles are concentrated. The circle diameters are proportional to the weights of the particles. 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 (i.e., the larger particles are repeated and the smaller particles are neglected). After resampling, the weights of all particles are same. Now these particles constitute the new set of particles. Then the whole procedure (Generic particle filter algorithm 2.4) is repeated with these new set of samples.

Although the resampling step reduces the effects of the degeneracy problem, it 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* [1,4]. There are techniques namely

*Markov chain Monte Carlo (MCMC)* [4], *regularization* [4] method to reduce the effect of sample impoverishment.

The sequential importance sampling algorithm is common for all types of particle filters. There are other versions of particle filters [2,4], namely (1) sampling importance resampling (SIR) filter; (2) auxiliary sampling importance resampling (ASIR) filter; (3)regularized particle filter (RPF).

### 2.4.3 Choice of Importance Density

The choice of the sampling density of the algorithm affects the quality of the state estimate significantly [2,4,12]. However there are number of choices for the sampling density. The sampling density must fulfil 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.

### Optimal Sampling Density

If 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 assume the form

$$q(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} = p(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$$

$$= \frac{p(\mathbf{z}_k, \mathbf{x}_k, \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k, \mathbf{x}_{k-1}^i)}$$

$$= \frac{p(\mathbf{z}_k / \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k, \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k / \mathbf{x}_{k-1}^i) p(\mathbf{x}_{k-1}^i)}$$

$$q(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} = \frac{p(\mathbf{z}_k / \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k / \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k / \mathbf{x}_{k-1}^i)} \tag{2.39}$$

Substitute the equation (2.39) in equation (2.35) we get

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k / \mathbf{x}_{k-1}^i)$$

$$= w_{k-1}^i \int p(\mathbf{z}_k / \mathbf{x}_k^{'}) p(\mathbf{x}_k^{'} / \mathbf{x}_{k-1}^{'}) d\mathbf{x}_k^{'} \quad (2.40)$$

Interestingly, the weights do not depend on the current value of the state $\mathbf{x}_k^i$. The above chosen optimal density has two limitations. It requires sampling from the pdf $p(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ and the evolution of integral expression (2.40). Both of them cannot be done easily. When $\mathbf{x}_k$ belongs to a finite set, then the integral expression (2.40) become a sum, and sampling from the optimal importance density is possible.

### Gaussian Optimal Importance Function

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. Such a system is given by

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \quad (2.41)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \quad (2.42)$$

Where $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are mutually independent zero-mean white Gaussian sequences with covariances $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$, respectively. It can be shown that in this case, both the optimal importance density and $p(\mathbf{z}_k / \mathbf{x}_{k-1})$ are Gaussian, that is:

$$p(\mathbf{x}_k / \mathbf{x}_{k-1}, \mathbf{z}_k) = N(\mathbf{x}_k; \mathbf{a}_k; \Sigma_k) \quad (2.43)$$

$$p(\mathbf{z}_k / \mathbf{x}_{k-1}) = N(\mathbf{z}_k; \mathbf{b}_k; \mathbf{S}_k) \quad (2.44)$$

Where $\mathbf{a}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{\Sigma}_k \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{z}_k - \mathbf{b}_k)$ (2.45)

$$\mathbf{\Sigma}_k = \mathbf{Q}_{k-1} - \mathbf{Q}_{k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \mathbf{H}_k \mathbf{Q}_{k-1} \quad (2.46)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{2.47}$$

$$\mathbf{b}_k = \mathbf{H}_k \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) \tag{2.48}$$

**Proof:**

According to the state space model described by (2.41) and (2.42), we observe that

$$p(\mathbf{x}_k/\mathbf{x}_{k-1}) = N(\mathbf{x}_k; \mathbf{f}_{k-1}(\mathbf{x}_{k-1}); \mathbf{Q}_{k-1}) \tag{2.49}$$

$$p(\mathbf{z}_k/\mathbf{x}_k) = N(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k; \mathbf{R}_k) \tag{2.50}$$

From the Baye's update formula

$$p(\mathbf{x}_k/\mathbf{x}_{k-1}, \mathbf{z}_k) = \frac{p(\mathbf{z}_k, \mathbf{x}_k, \mathbf{x}_{k-1})}{p(\mathbf{z}_k, \mathbf{x}_{k-1})}$$

$$= \frac{p(\mathbf{z}_k/\mathbf{x}_k, \mathbf{x}_{k-1}) p(\mathbf{x}_k, \mathbf{x}_{k-1})}{p(\mathbf{z}_k/\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1})}$$

$$= \frac{p(\mathbf{z}_k/\mathbf{x}_k) p(\mathbf{x}_k/\mathbf{x}_{k-1})}{p(\mathbf{z}_k/\mathbf{x}_{k-1})}$$

$$p(\mathbf{x}_k/\mathbf{x}_{k-1}, \mathbf{z}_k) p(\mathbf{z}_k/\mathbf{x}_{k-1}) = p(\mathbf{z}_k/\mathbf{x}_k) p(\mathbf{x}_k/\mathbf{x}_{k-1}) \tag{2.51}$$

By taking the exponent terms on the R.H.S of the expression (2.51), we get

$$\Rightarrow (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k)^T \mathbf{R}_k^{-1} (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k) + (\mathbf{x}_k - \mathbf{f}_{k-1}(\mathbf{x}_{k-1}))^T \mathbf{Q}_{k-1}^{-1} (\mathbf{x}_k - \mathbf{f}_{k-1}(\mathbf{x}_{k-1}))$$

$$\Rightarrow \mathbf{z}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k - \mathbf{z}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{x}_k - \mathbf{x}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k + \mathbf{x}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{x}_k + \mathbf{x}_k^T \mathbf{Q}_{k-1}^{-1} \mathbf{x}_k - \mathbf{x}_k^T \mathbf{Q}_{k-1}^{-1} \mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

$$-\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1}) \mathbf{Q}_{k-1}^{-1} \mathbf{x}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1}) \mathbf{Q}_{k-1}^{-1} \mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

$$\Rightarrow \mathbf{x}_k^T \left[\mathbf{Q}_{k-1}^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k\right] \mathbf{x}_k - \mathbf{x}_k^T \left[\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k + \mathbf{Q}_{k-1}^{-1} \mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$-\left[\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\right]\mathbf{x}_k+\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) \tag{2.52}$$

By matrix inversion lemma, we have

$$\left[\mathbf{Q}_{k-1}^{-1}+\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right]^{-1}=\mathbf{Q}_{k-1}-\mathbf{Q}_{k-1}\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{Q}_{k-1}\mathbf{H}_k^T+\mathbf{R}_k\right]^{-1}\mathbf{H}_k\mathbf{Q}_{k-1}$$

Let $\quad \mathbf{\Sigma}_k^{-1}=\mathbf{Q}_{k-1}^{-1}+\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k$ and $\mathbf{S}_k=\mathbf{H}_k\mathbf{Q}_{k-1}\mathbf{H}_k^T+\mathbf{R}_k$

$$\Rightarrow \mathbf{\Sigma}_k=\mathbf{Q}_{k-1}-\mathbf{Q}_{k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}\mathbf{H}_k\mathbf{Q}_{k-1}$$

The term $-\mathbf{x}_k^T\left[\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k+\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$ can be simplified as

$$\Rightarrow -\mathbf{x}_k^T\left[\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k+\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})+\mathbf{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})-\mathbf{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow -\mathbf{x}_k^T\left[\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k+\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})+\mathbf{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})-\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})-\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow -\mathbf{x}_k^T\left[\mathbf{H}_k^T\mathbf{R}_k^{-1}\left\{\mathbf{z}_k-\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right\}+\mathbf{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

Let $\mathbf{b}_k=\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$ and $\mathbf{a}_k=\mathbf{f}_{k-1}(\mathbf{x}_{k-1})+\mathbf{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}(\mathbf{z}_k-\mathbf{b}_k)$ then

$$\mathbf{\Sigma}_k^{-1}\left[\mathbf{a}_k-\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]=\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k-\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow -\mathbf{x}_k^T\left[\mathbf{\Sigma}_k^{-1}\left[\mathbf{a}_k-\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]+\mathbf{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow -\mathbf{x}_k^T\mathbf{\Sigma}_k^{-1}\mathbf{a}_k \tag{2.53}$$

The term $-\left[\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\right]\mathbf{x}_k$ can be simplified as

$$\Rightarrow -\left[\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{\Sigma}_k^{-1}-\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{\Sigma}_k^{-1}\right]\mathbf{x}_k$$

$$\Rightarrow -\left[\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{\Sigma}_k^{-1}-\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\left[\mathbf{Q}_{k-1}^{-1}+\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right]\right]\mathbf{x}_k$$

$$\Rightarrow -\left[\left[\mathbf{z}_k^T - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\right]\mathbf{R}_k^{-1}\mathbf{H}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\boldsymbol{\Sigma}_k^{-1}\right]\mathbf{x}_k$$

$$\Rightarrow -\left[\left[\mathbf{a}_k^T - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\right]\boldsymbol{\Sigma}_k^{-1} + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\boldsymbol{\Sigma}_k^{-1}\right]\mathbf{x}_k$$

$$\Rightarrow -\mathbf{a}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_k \tag{2.54}$$

By using equations (2.53) and (2.54), the equation (2.52) can be further simplified as

$$\Rightarrow (\mathbf{x}_k - \mathbf{a}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_k - \mathbf{a}_k) + \mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k - \mathbf{a}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{a}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) \tag{2.55}$$

The term $\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k - \mathbf{a}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{a}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$ can be simplified as

$$\Rightarrow -\left[\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1}) + \left[\mathbf{z}_k^T - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\right]\mathbf{R}_k^{-1}\mathbf{H}_k\boldsymbol{\Sigma}_k\right]\boldsymbol{\Sigma}_k^{-1}\left[\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \boldsymbol{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]\right]$$
$$+\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

$$\Rightarrow -\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\boldsymbol{\Sigma}_k^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$-\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\mathbf{H}_k\boldsymbol{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$-\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

$$\Rightarrow -\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\mathbf{H}_k\boldsymbol{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$-\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k$$

$$+\mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\left[\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k + \mathbf{Q}_{k-1}^{-1}\right]\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

$$+\mathbf{z}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k + \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{Q}_{k-1}^{-1}\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$

By matrix inversion lemma we have $\mathbf{S}_k^{-1} = \mathbf{R}_k^{-1} - \mathbf{R}_k^{-1}\mathbf{H}_k\boldsymbol{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}$, now

$$\Rightarrow -\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right] + \left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{S}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$
$$+\mathbf{z}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right] - \mathbf{f}_{k-1}^T(\mathbf{x}_{k-1})\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow -\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right] + \left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{S}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$
$$+\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{R}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow \left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]^T \mathbf{S}_k^{-1}\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{f}_{k-1}(\mathbf{x}_{k-1})\right]$$

$$\Rightarrow \left[\mathbf{z}_k - \mathbf{b}_k\right]^T \mathbf{S}_k^{-1}\left[\mathbf{z}_k - \mathbf{b}_k\right] \tag{2.56}$$

By substituting equation (2.56) in the equation (2.55) then

$$\Rightarrow (\mathbf{x}_k - \mathbf{a}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}_k - \mathbf{a}_k) + \left[\mathbf{z}_k - \mathbf{b}_k\right]^T \mathbf{S}_k^{-1}\left[\mathbf{z}_k - \mathbf{b}_k\right]$$

$$\Rightarrow p(\mathbf{x}_k/\mathbf{x}_{k-1},\mathbf{z}_k)\,p(\mathbf{z}_k/\mathbf{x}_{k-1}) = N(\mathbf{x}_k;\mathbf{a}_k;\mathbf{\Sigma}_k)\,N(\mathbf{z}_k;\mathbf{b}_k;\mathbf{S}_k)$$

Hence it is proved. The analytical expressions obtained above are difficult for most other cases. In the following, the suboptimal choices of the sampling density will be considered.

### Prior Sampling Density

This sampling density is frequently used due to its simplicity and easy weight computation. Here the current estimate $\mathbf{z}_k$ is ignored during drawing of samples and thus low quality estimates will be obtained. The prior sampling density takes the form [2,4] as

$$q(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k / \mathbf{x}_{k-1}^i) \tag{2.57}$$

By substituting the equation (2.57) in the equation (2.35) we get

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k / \mathbf{x}_k^i) \tag{2.58}$$

The equation (2.40) states that it is possible to calculate the importance weights before the particles are propagated to time $k$. The equation (2.58) states that this is not possible with the prior sampling density.

If the transitional prior $p(\mathbf{x}_k/\mathbf{x}_{k-1})$ is used as the importance density and is a much broader distribution than the likelihood, $p(\mathbf{z}_k/\mathbf{x}_k)$, 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.

## 2.5 Sampling Importance Resampling (SIR) Filter

The SIR algorithm in can be easily obtained from SIS algorithm by considering the following [2].

- The importance density is chosen to be prior density

$$q(\mathbf{x}_k / \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k / \mathbf{x}_{k-1}^i)$$

- The resampling step is carried out at every time index.

By the choice of importance density as the prior density, the weights are given by

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k / \mathbf{x}_k^i)$$

However, considering the fact that the resampling step is carried out at every time index, the weight update is given by

$$w_k^i \propto p(\mathbf{z}_k / \mathbf{x}_k^i) \tag{2.59}$$

The weights should be normalized before resampling step is carried out. The advantage of SIR over SIS is easy weight computation. The SIR filter algorithm 2.5 is presented in the following [2,4].

*Algorithm 2.7: SIR Particle filter*

$$\left[\left\{\mathbf{x}_k^i, w_k^i\right\}_{i=1}^{N_s}\right] = SIR\left[\left\{\mathbf{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^{N_s}, \mathbf{z}_k\right]$$

- *FOR* $i=1:N_s$

  - *Draw* $\mathbf{x}_k^i \sim p(\mathbf{x}_k / \mathbf{x}_{k-1}^i)$
  - *Assign each particle with the importance weight up to a normalizing constant according to*

    $$\tilde{w}_k^i = p(\mathbf{z}_k / \mathbf{x}_k^i)$$

- *END FOR*
- *Calculate the total weight:* $t = SUM\left[\left\{\tilde{w}_k^i\right\}_{i=1}^{N_s}\right]$
- *FOR* $i=1:N_s$

  - *Normalize the weights:* $w_k^i = t^{-1}\tilde{w}_k^i$

- *END FOR*
- *Resample using systematic resampling or residual resampling.*

## 2.6 Simulation Results

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

$$x_k = f_k\left(x_{k-1}, k\right) + v_{k-1} \tag{2.60}$$

$$z_k = \frac{x_k^2}{20} + n_k \tag{2.61}$$

where

$$f_k\left(x_{k-1}, k\right) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos\left(1.2k\right) \tag{2.62}$$

From the state space model (2.60) & (2.61), the prior density $p(x_k/x_{k-1})$ and likelihood function $p(z_k/x_k)$ are respectively given by

$$p(x_k/x_{k-1}) = N(x_k; f_k(x_{k-1},k), Q_{k-1}) \tag{2.63}$$

$$p(z_k/x_k) = N\left(z_k; \frac{x_k^2}{20}, R_k\right) \tag{2.64}$$

It is assumed that in equations (2.60) & (2.61), $v_{k-1}$ and $n_k$ 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 $Q_{k-1} = 10$ and $R_k = 1$ respectively.
- Number of states $M$=100
- Number of particles $N$=10,100
- Number of Monte Carlo runs=1000

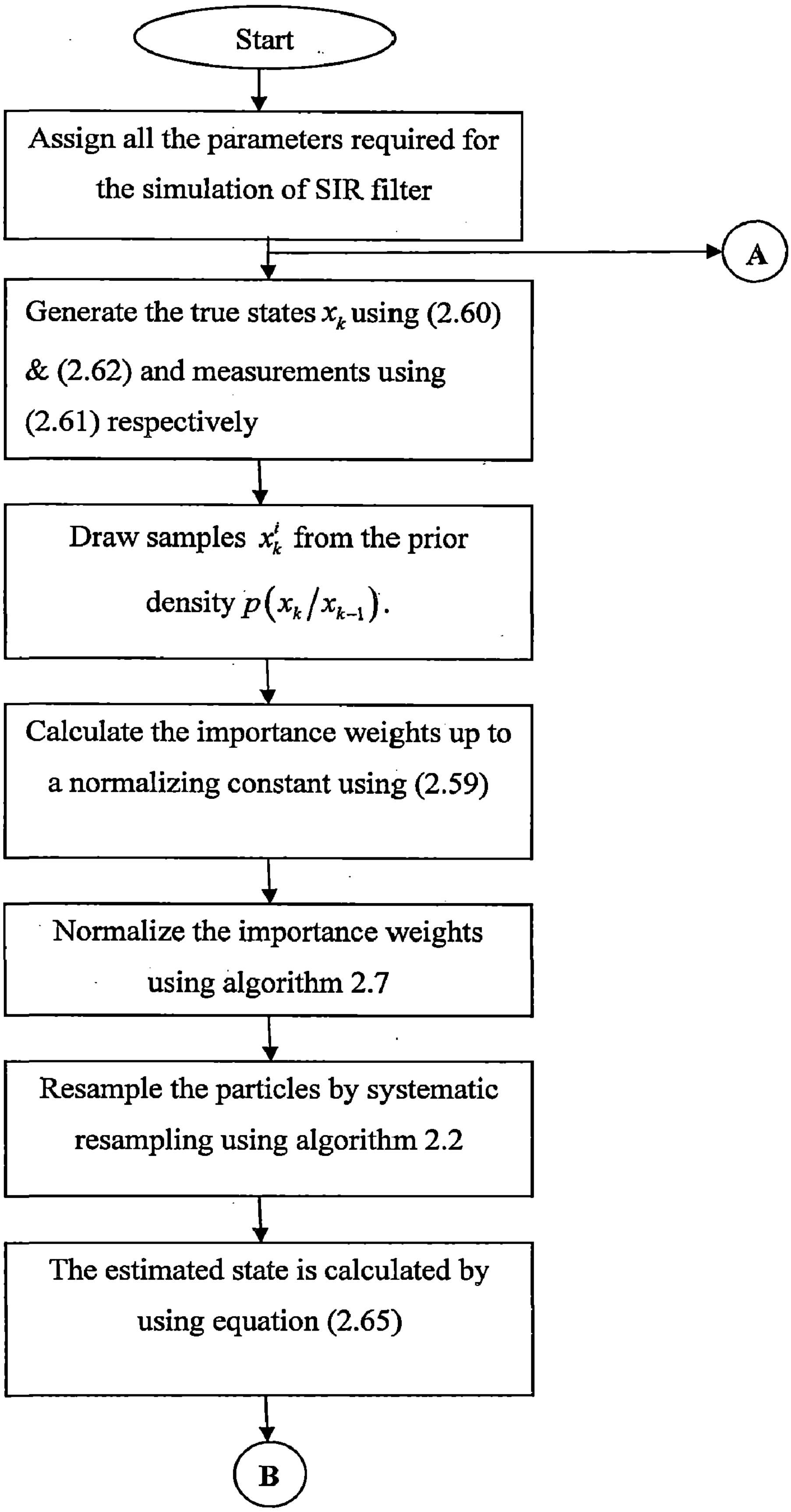
The samples $\{x_k^i\}_{i=1}^N$ and the corresponding weights $\{w_k^i\}_{i=1}^N$ are generated using algorithm 2.7. The estimate of the state $x_{est}$ is calculated by using the set of samples $\{x_k^i\}_{i=1}^N$ and corresponding weights $\{w_k^i\}_{i=1}^N$, which is given by the sum of products of samples and corresponding weights.

$$x_{est} = \sum_{i=1}^{N} x_k^i w_k^i \tag{2.65}$$

To obtain the performance of state estimation, the Root Mean Square Error (RMSE) between the true state and estimated state is computed, which is given by

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x - x_{est})^2} \tag{2.66}$$

The flow chart for simulation of SIR particle filter algorithm is given in Fig.2.6.



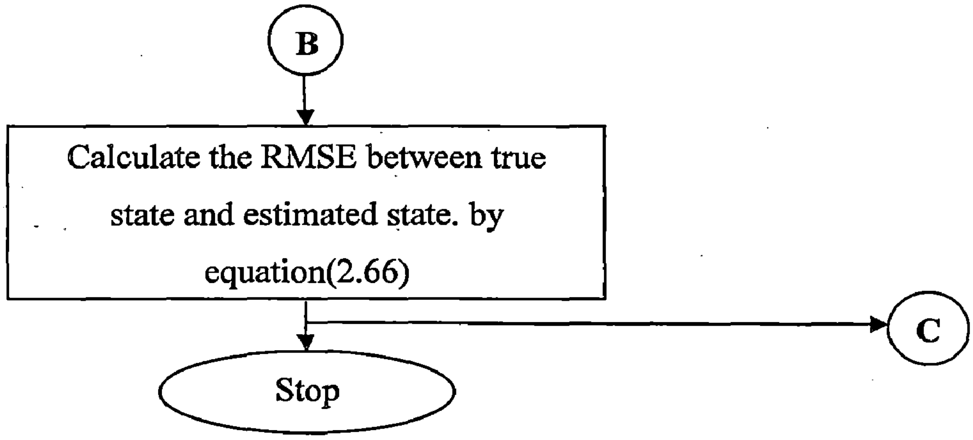



**Figure 2.6** Flow chart for the simulation of SIR particle filter

All steps from A to C shown in flow chart are repeated for each independent Monte Carlo run. Fig 2.7 shows 100 true values of the state $x_k$ as a function of time $k$. Fig 2.8 shows the 100 measurements $z_k$ of the state $x_k$ as a function of time $k$.

Fig 2.9 shows the estimated state and true state for comparison. In this case, SIR filter uses 10 particles for estimating the state. The RMSE of SIR filter is obtained by averaging over 1000 independent realizations, which is found to be 16.6144.

Fig 2.10 shows the estimated state of the SIR filter when 100 particles are used. For comparison, we have also plotted the true states $x_k$. 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.

It can be seen from the Fig2.9, the SIR filter gives disappointing results when 10 particles are used. It is observed that there is nearly 3 times improvement in the RMSE when 100 particles are used. So, to achieve smaller errors, we have to increase the number of particles. The advantage with smaller number of particles is that lower numbers of computations are needed.

In this chapter, we have considered the application of particle filtering to estimate the state of a nonlinear system. In following chapters, we will consider that application of particle filtering for blind detection in SISO, MIMO, OFDM and MIMO-OFDM systems.

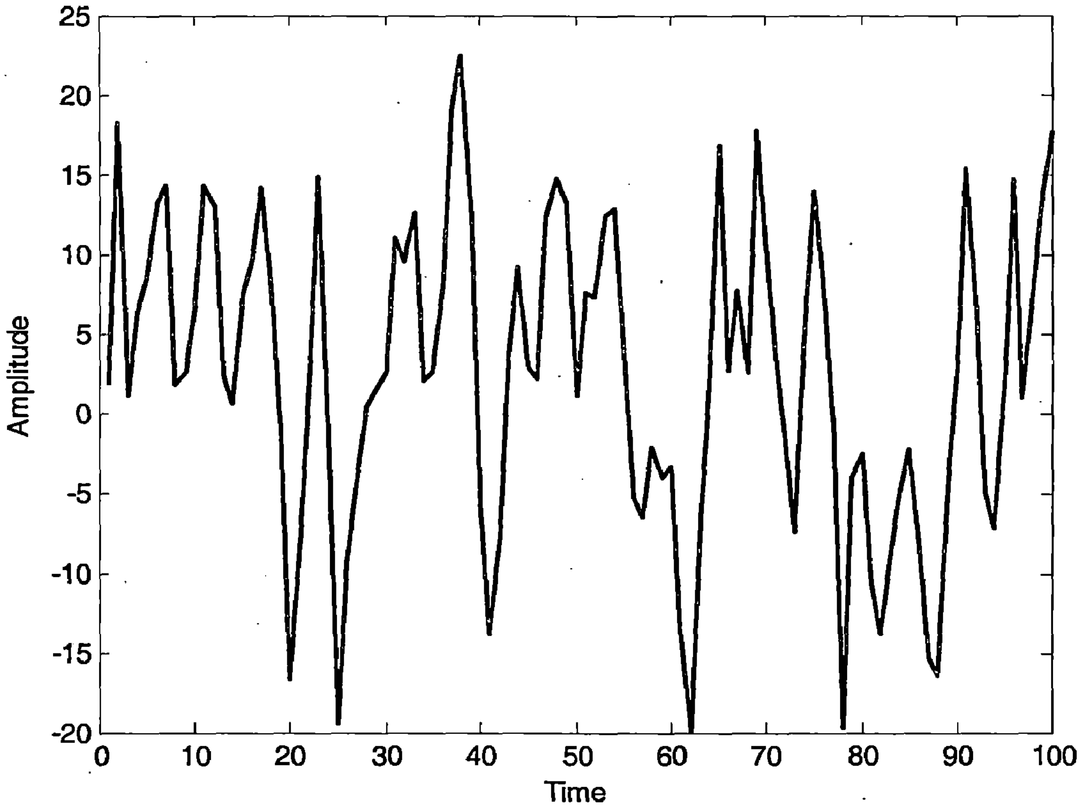



**Figure 2.7** 100 true values of the state $x_k$ as a function of time $k$

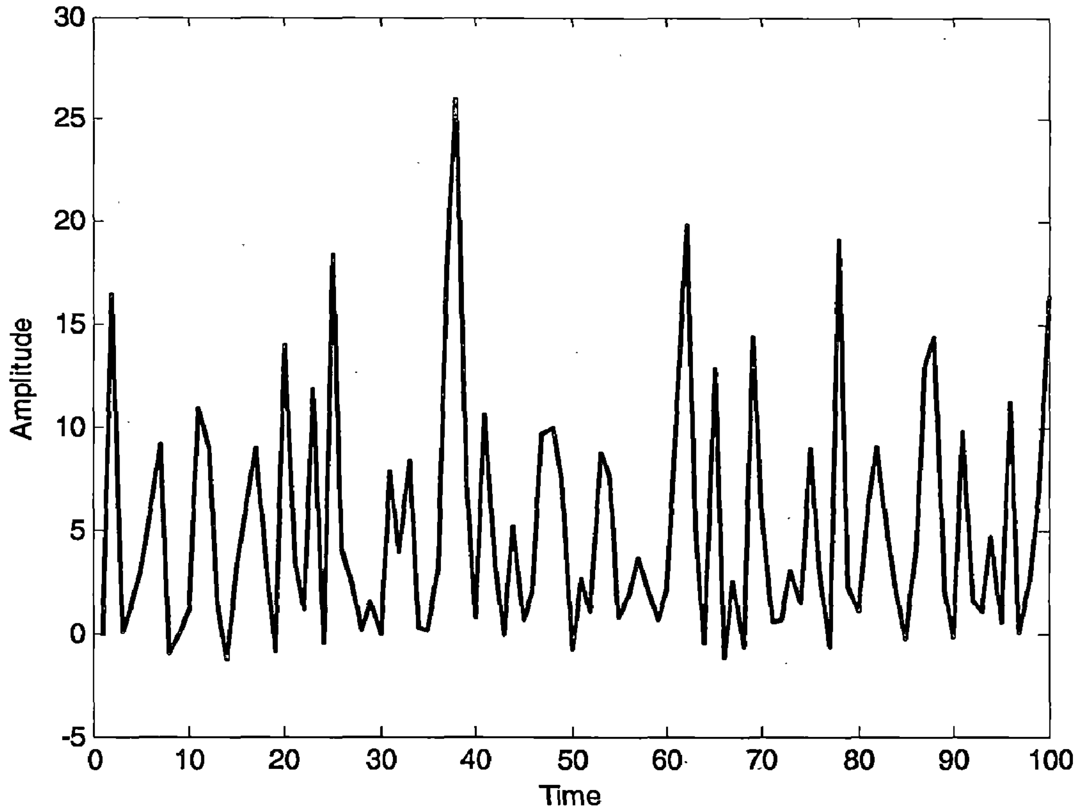



**Figure 2.8** 100 measurements $z_k$ of the state $x_k$ as a function of time $k$

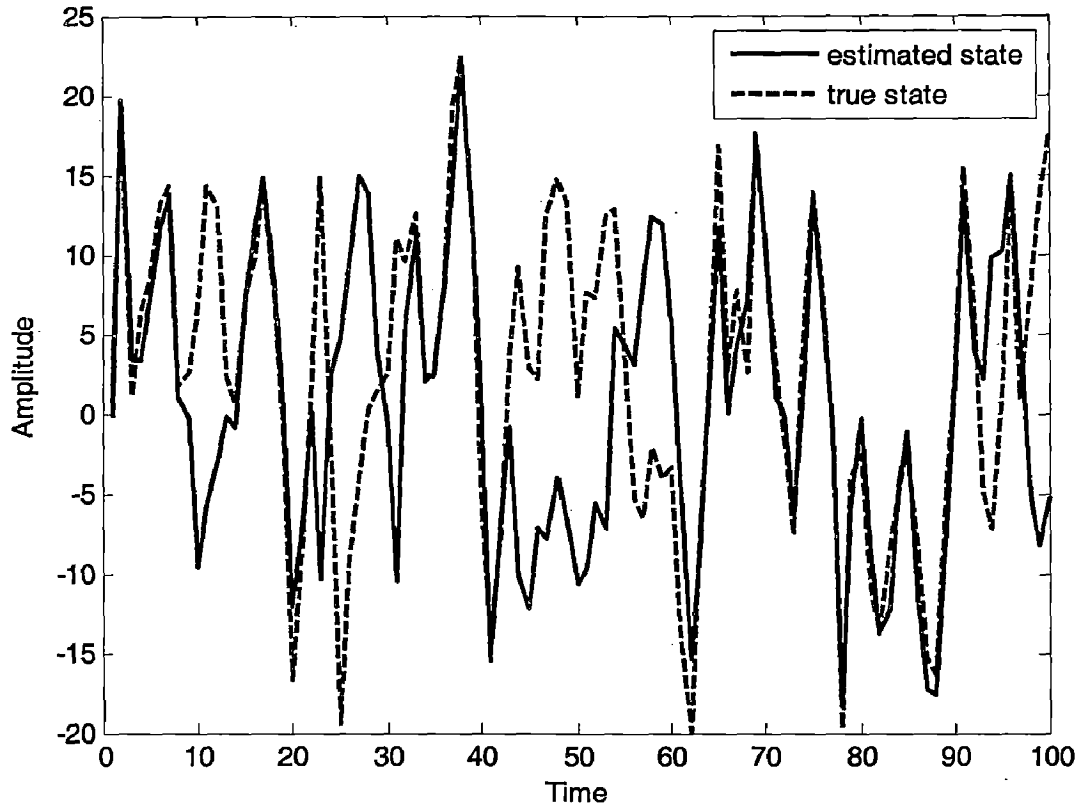



**Figure 2.9** True and estimated values of the state $x_k$ as a function of time $k$ considering 10 particles

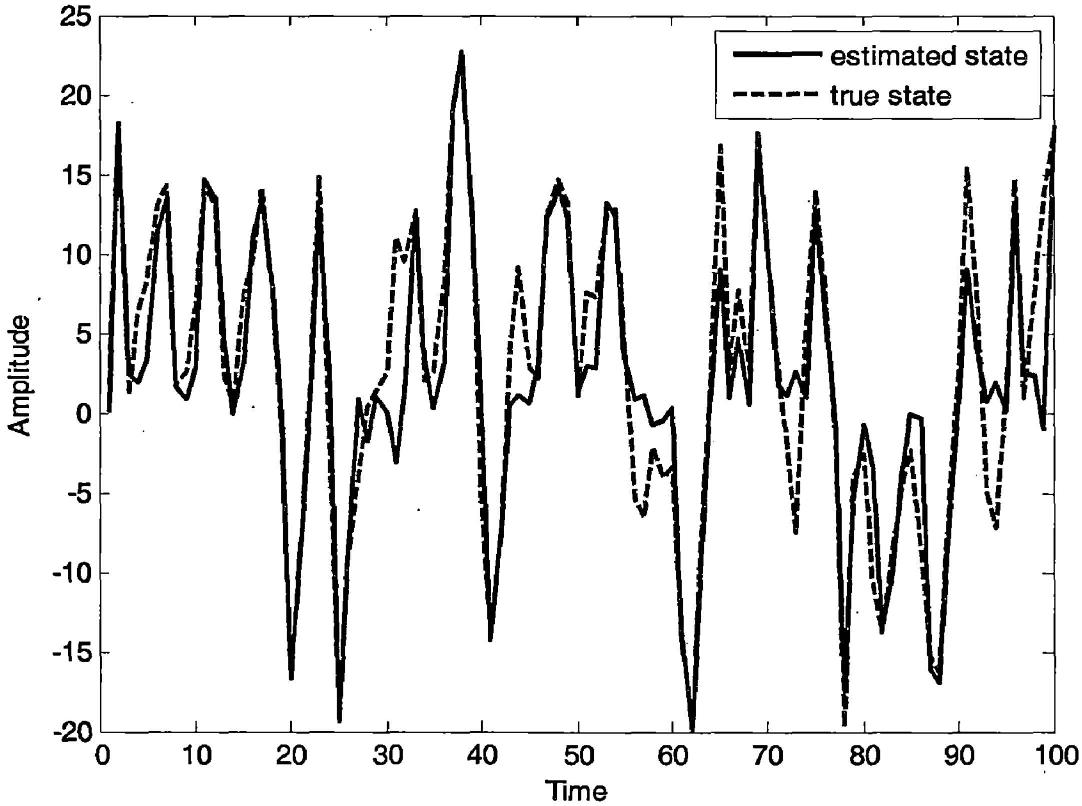



**Figure 2.10** True and estimated values of the state $x_k$ as function of time $k$ considering 100 particles

# Chapter 3

# PARTICLE FILTERING FOR BLIND DETECTION IN SISO AND MIMO SYSTEMS

In the transmission of digital information over a communication channel which is fading dispersive, caused by the interference between two or more versions of transmitted signal which arrive at the receiver at slightly different times [14,15], and for the case of known channels the optimal detection is performed by the maximum-likelihood sequence estimation (MLSE) detector [14,30]. 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. Zero-forcing (ZF) detector [14] and the minimum mean square error (MMSE) detector [14] require only linear complexity, but cannot achieve optimal performance. 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[31] suggested a pilot method for detection of signals in fading channels. But the transmission of pilot requires bandwidth, decreases the communication throughput and causes significant overhead problem. This loss is insignificant for time invariant channel where as in case of time varying channel, the loss is significant.

A novel adaptive Bayesian receiver for signal detection and decoding in fading channels with unknown channel statistics is presented in [16]. 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 [16]. We have used the state space model approach for deriving the particle filtering algorithm for the blind detection in single-input single-output (SISO) systems with the use of Kalman filtering algorithm. 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, SISO communication system is described and 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 first. The derivation of particle filter algorithm for signal detection in fading channels for SISO systems is presented next. The residual resampling algorithm and the delayed estimation approach are also discussed. Finally above approach is used for blind detection in MIMO system. Simulation results are given at the end.

## 3.1 Signal Model of SISO System

Consider a communication system signalling through a flat fading channel with additive ambient noise as given in Fig.3.1 [16,17].

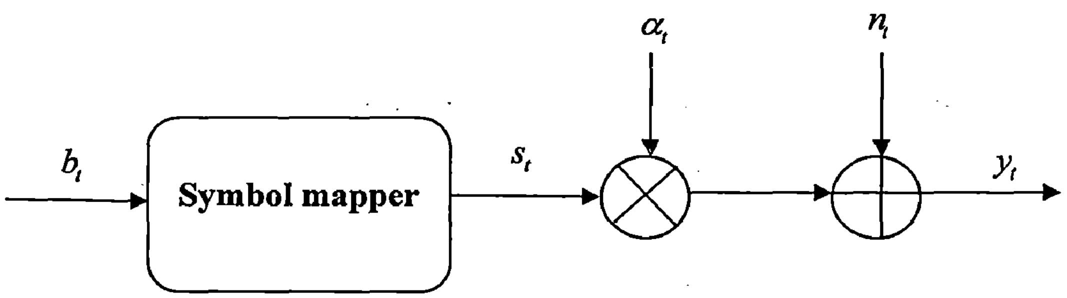



**Figure 3.1** Communication system signalling through flat fading channel

As Fig 3.1 shows, the input binary 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 $A = \{a_1, \ldots\ldots, a_{|A|}\}$. 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 [16,17]

$$y_t = \alpha_t s_t + n_t, \qquad t=0,1,\ldots\ldots \tag{3.1}$$

Where $\alpha_t$ is the fading channel coefficient

$s_t$ is the transmitted symbol

$n_t$ is the ambient additive channel noise at time *t*.

The processes $\{\alpha_t\}$, $\{s_t\}$, $\{n_t\}$ are assumed to be mutually independent. It is assumed that the additive noise $n_t$ in equation (3.1) is a sequence of independent and identically distributed (i.i.d) zero-mean complex random variables. Two types of noise distributions are considered. In the first type, $n_t$ assumes a complex Gaussian distribution of zero mean and variance $\sigma^2$, which is given by

$$n_t \sim N_c\left(0,\sigma^2\right) \tag{3.2}$$

Where as in second type, $n_t$ assumes Middeleton Class A noise model [9,32] for modelling a non-Gaussian distribution, which has been extensively used to model physical noise in radio and acoustic channels. i.e., $n_t$ takes the form of a two-term mixture Gaussian distribution [9].

$$n_t \sim (1-\varepsilon)N_c\left(0,\varsigma^2\right)+\varepsilon N_c\left(0,k\varsigma^2\right) \tag{3.3}$$

where $N_c\left(0,\varsigma^2\right)$ represents the nominal ambient noise.

$N_c\left(0,k\varsigma^2\right)$ represents an impulsive component

$\varepsilon$ is the probability that impulsive pulses can occur, $0<\varepsilon<1$

$k>1$.

The overall variance of the noise is fixed by varying the parameters $\varepsilon$ and $k$, which is given by [9]

$$\sigma^2=(1-\varepsilon)\varsigma^2+k\varsigma^2 \tag{3.4}$$

It is further assumed that the channel-fading process is Rayleigh i.e., the fading coefficients $\{\alpha_t\}$ form a complex Gaussian process [16]. The fading process is usually modelled by the output of a Butterworth filter driven by white Gaussian noise.

### 3.1.1 Fading Coefficients Modelled as ARMA Process

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

$$\phi_r\alpha_{t-r} + \ldots\ldots\ldots\ldots + \phi_1\alpha_{t-1} + \alpha_t = \theta_0 u_t + \theta_1 u_{t-1} + \ldots\ldots\ldots + \theta_r u_{t-r} \tag{3.5}$$

where $\{u_t\}$ is a white complex Gaussian noise sequence with independent real and complex components. The ARMA coefficients $\{\phi_i\}$ and $\{\theta_i\}$, as well as 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[16]. 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 $x_t$, which is given by

$$x_t = -\phi_1 x_{t-1} - \phi_2 x_{t-2} \ldots\ldots\ldots\ldots - \phi_r x_{t-r} + u_t \tag{3.6}$$

By writing the equation (3.6) in matrix form, we get

$$\begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-r} \end{bmatrix} = \begin{bmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \\ \vdots \\ x_{t-r-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_t \tag{3.7}$$

Denote $\mathbf{x}_t = \begin{bmatrix} x_t & x_{t-1} & x_{t-2} & \cdots & x_{t-r} \end{bmatrix}^T$ then state equation is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{3.8}$$

Where $\mathbf{F} = \begin{bmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$

$$g = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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

Now, from the equation (3.5) & (3.6), we get

$$\alpha_t = \theta_0 x_t + \theta_1 x_{t-1} + \dots\dots\dots + \theta_r x_{t-r} \tag{3.9}$$

By writing equation (3.9) in matrix form, we get

$$\alpha_t = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \cdots & \theta_r \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-r} \end{bmatrix} \tag{3.10}$$

Denote $\mathbf{h} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \cdots & \theta_r \end{bmatrix}^H$ then $\alpha_t$ is given by

$$\alpha_t = \mathbf{h}^H \mathbf{x}_t \, . \tag{3.11}$$

If the additive noise in equation (3.1) is Gaussian i.e., $n_t \sim N_c(0, \sigma^2)$, then the state-space model for SISO system is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{3.12}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t \tag{3.13}$$

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

If the additive noise in equation (3.1) is non-Gaussian and is modeled by equation (3.3), then an indicator random variable $I_t$, t=0,1..... is used to model the state space model. The indicator variable is defined by

$$I_t \triangleq \begin{cases} 1, & \text{if } n_t \sim N_c\left(0,\varsigma^2\right) \\ 2 & \text{if } n_t \sim N_c\left(0,k\varsigma^2\right) \end{cases} \tag{3.14}$$

with $p(I_t = 1) = (1-\varepsilon)$ and $p(I_t = 2) = \varepsilon$ . Let $\sigma_1 = \varsigma^2$ and $\sigma_2 = k\varsigma^2$ , then state space model of the system for the case of non-Gaussian noise is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{3.15}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma_{I_t} v_t \tag{3.16}$$

### 3.1.2 Fading Coefficients Modelled as AR Process

The generalized form of AR process of order $r$ is given by

$$\phi_r \alpha_{t-r} + \ldots\ldots\ldots\ldots + \phi_1 \alpha_{t-1} + \alpha_t = u_t \tag{3.17}$$

where $\{u_t\}$ is a white complex Gaussian noise sequence with independent real and complex components and $\{\phi_i\}$ are AR coefficients.

By defining state variable $x_t$ as in equation (3.6) and $\mathbf{x}_t = \begin{bmatrix} x_t & x_{t-1} & x_{t-2} & \cdots & x_{t-r} \end{bmatrix}^T$ then state equation is given by

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

Where $\mathbf{F} = \begin{bmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$

$$g = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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

Now form equation (3.6) & (3.17), we get

$$\alpha_t = x_t \tag{3.18}$$

By writing the equation (3.18) in matrix form as

$$\alpha_t = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-r} \end{bmatrix} \tag{3.19}$$

Denote $\mathbf{h} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}^H$ then $\alpha_t$ is given by

$$\alpha_t = \mathbf{h}^H \mathbf{x}_t \tag{3.20}$$

If the additive noise is Gaussian i.e., $n_t \sim N_c\left(0, \sigma^2\right)$, then the state-space model for the SISO system is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{3.21}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t \tag{3.22}$$

On the other hand, if the additive noise is non-Gaussian, then state space model of the SISO system is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t \tag{3.23}$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma_{I_t} v_t \tag{3.24}$$

## 3.2 Particle Filtering Algorithm for SISO System

Consider the flat-fading channel with additive Gaussian noise given by (3.12) & (3.13). Let $\mathbf{Y}_t \triangleq (y_0, \ldots\ldots\ldots, y_t)$ be the received signals and $\mathbf{S}_t \triangleq (s_0, \ldots\ldots\ldots, s_t)$ be the transmitted signals up to time *t* respectively.

***Statement of the Problem:*** *To estimate the a posteriori probabilities of the information symbols*

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

*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.,

$$a_i = exp\left(j\frac{2\pi i}{|A|}\right), \qquad for \;\; i = 0, \ldots\ldots\ldots, |A|-1 \tag{3.25}$$

where $j = \sqrt{-1}$. Assume that the transmitted symbols are independent i.e.,

$$p(s_t = a_i / \mathbf{S}_{t-1}) = p(s_t = a_i), \qquad a_i \in A \tag{3.26}$$

When no prior information about the symbols is available, the symbols are assumed to take each possible value in the Alphabet set $A$ with equal probability i.e.,

$$p(s_t = a_i) = \frac{1}{|A|} \quad \text{for} \;\; i = 1, \ldots\ldots\ldots |A|. \tag{3.27}$$

In order to implement the particle filter, a set of Monte Carlo samples of the transmitted symbols $\left\{\mathbf{S}_t^{(j)}\right\}_{j=1}^{m}$ with its corresponding importance weights $\left\{w_t^{(j)}\right\}_{j=1}^{m}$ which are properly weighted with respect to the distribution $p(\mathbf{S}_t / \mathbf{Y}_t)$ are needed. From SIS discussed in section 2.4.1, the Monte Carlo samples are easily generated from a trial sampling density. If the choice of trial sampling density is taken as the optimal sampling density then by equation (2.39)

$$q\left(s_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_t\right)=p\left(s_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_t\right) \tag{3.28}$$

For this choice of sampling density, the weights are updated according to equation (2.40) as

$$w_t^{(j)} \propto w_{t-1}^{(j)} p\left(y_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right) \tag{3.29}$$

$$w_t^{(j)} \propto w_{t-1}^{(j)} \sum_{a_i \in A} p\left(y_t/s_t=a_i,\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)p\left(s_t=a_i\right) \tag{3.30}$$

Denote $\rho_{t,i}^{(j)}=p\left(y_t/s_t=a_i,\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)p(s_t=a_i)$, then weights are given by

$$w_t^{(j)} \propto w_{t-1}^{(j)} \sum_{a_i \in A} \rho_{t,i}^{(j)} \tag{3.31}$$

Now, the term sampling density $p\left(s_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_t\right)$ is modified as

$$\begin{aligned}
p\left(s_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_t\right) &= p\left(s_t/\mathbf{S}_{t-1}^{(j)},y_t,\mathbf{Y}_{t-1}\right) \\
&= \frac{p\left(s_t,\mathbf{S}_{t-1}^{(j)},y_t,\mathbf{Y}_{t-1}\right)}{p\left(\mathbf{S}_{t-1}^{(j)},y_t,\mathbf{Y}_{t-1}\right)} \\
&= \frac{p\left(y_t/\mathbf{S}_{t-1}^{(j)},s_t,\mathbf{Y}_{t-1}\right)p\left(\mathbf{S}_{t-1}^{(j)},s_t,\mathbf{Y}_{t-1}\right)}{p\left(y_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)p\left(\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)} \\
&= \frac{p\left(y_t/s_t,\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)p\left(s_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)}{p\left(y_t/\mathbf{S}_{t-1}^{(j)},\mathbf{Y}_{t-1}\right)}
\end{aligned}$$

$$\propto p\left(y_t / s_t, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) p\left(s_t / \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)$$

$$\propto p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) p(s_t = a_i)$$

$$p\left(s_t / \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_t\right) = \rho_{t,i}^{(j)} \tag{3.32}$$

From the state space model (3.12) & (3.13), the density $p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)$ is Gaussian and its mean and variance is calculated using the Kalman filtering algorithm.

$$p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \sim N_c\left(mean, variance\right) \tag{3.33}$$

The state space model defined by equations (3.12) & (3.13) is reproduced for convenience.

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

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

The Kalman filtering algorithm for the above state space model when $s_t$ is known, is given below:

The innovation term from equation (2.17) is given by

$$\eta_t = y_t - s_t \mathbf{h}^H \mathbf{F} \hat{\mathbf{x}}_{t-1} \tag{3.34}$$

The correlation matrix of the innovation process by using equation (2.19) is given by

$$R_t = s_t \mathbf{h}^H \mathbf{K}_t \mathbf{h} s_t^* + \sigma^2$$

$$R_t = \left|s_t\right|^2 \mathbf{h}^H \mathbf{K}_t \mathbf{h} + \sigma^2$$

$$R_t = \mathbf{h}^H \mathbf{K}_t \mathbf{h} + \sigma^2 \qquad (\because \left|s_t\right| = 1) \tag{3.35}$$

The Kalman gain by using equation (2.20) is given by

$$\mathbf{g}_t = \mathbf{K}_t \mathbf{h} s_t^* / R_t \tag{3.36}$$

Filtered state error correlation matrix by using equation (2.18) is given by

$$\mathbf{\Sigma}_t = \mathbf{K}_t - \mathbf{g}_t s_t \mathbf{h}^H \mathbf{K}_t$$

$$\mathbf{\Sigma}_t = \mathbf{K}_t - \frac{1}{R_t} \mathbf{K}_t \mathbf{h} s_t^* s_t \mathbf{h}^H \mathbf{K}_t$$

$$\mathbf{\Sigma}_t = \mathbf{K}_t - \frac{1}{R_t} |s_t|^2 \mathbf{K}_t \mathbf{h}\mathbf{h}^H \mathbf{K}_t$$

$$\mathbf{\Sigma}_t = \mathbf{K}_t - \frac{1}{R_t} \mathbf{K}_t \mathbf{h}\mathbf{h}^H \mathbf{K}_t \qquad (\because |s_t| = 1) \tag{3.37}$$

Predicted state error correlation matrix by using equation (2.16) is given by

$$\mathbf{K}_t = \mathbf{F} \mathbf{\Sigma}_{t-1} \mathbf{F}^H + \mathbf{g}\mathbf{g}^H \tag{3.38}$$

Estimated state vector by using equation (2.17) is given by

$$\hat{\mathbf{x}}_t = \mathbf{F} \hat{\mathbf{x}}_{t-1} + \mathbf{g}_t \eta_t$$

$$\hat{\mathbf{x}}_t = \mathbf{F} \hat{\mathbf{x}}_{t-1} + \frac{1}{R_t} \mathbf{K}_t \mathbf{h} s_t^* \left( y_t - s_t \mathbf{h}^H \mathbf{F} \hat{\mathbf{x}}_{t-1} \right)$$

$$\hat{\mathbf{x}}_t = \mathbf{F} \hat{\mathbf{x}}_{t-1} + \frac{1}{R_t} \left( y_t - s_t \mathbf{h}^H \mathbf{F} \hat{\mathbf{x}}_{t-1} \right) s_t^* \mathbf{K}_t \mathbf{h} \tag{3.39}$$

The *m ean* of density $p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)$ can be computed using the innovation equation (3.34) as [16]

$$mean = E\left\{ y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1} \right\} \tag{3.40}$$

$$= s_t \mathbf{h}^H \mathbf{F} \mathbf{x}_{t-1}^{(j)} \Big|_{s_t = a_i}$$

$$= a_i \mathbf{h}^H \mathbf{F} \mathbf{x}_{t-1}^{(j)}$$

Now denote $\mu_t^{(j)} = \mathbf{h}^H \mathbf{F} \mathbf{x}_{t-1}^{(j)}$, then

$$mean = a_i \mu_t^{(j)} \tag{3.41}$$

The *variance* of density $p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)$ can be computed by using the correlation matrix of the innovation process equation (3.35) as [16]

$$variance = \text{Var}\left\{y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right\} \tag{3.42}$$

$$= \mathbf{h}^H \mathbf{K}_t^{(j)} \mathbf{h} + \sigma^2 \Big|_{s_t = a_i}$$

$$= \mathbf{h}^H \mathbf{K}_t^{(j)} \mathbf{h} + \sigma^2$$

By using the equation (3.35), *variance* is given by

$$variance = R_t^{(j)} \tag{3.43}$$

Now, the probability density $p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)$ is given by

$$p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \sim N_c\left(a_i \mu_t^{(j)}, R_t^{(j)}\right) \tag{3.44}$$

For each $a_i \in A$, the *a posteriori* symbol probability $p\left(s_t = a_i / \mathbf{Y}_t\right)$ as in equation (2.36) can be estimated as [5,9]

$$p\left(s_t = a_i / \mathbf{Y}_t\right) = E\left\{\delta\left(s_t = a_i\right) / \mathbf{Y}_t\right\}$$

$$\cong \frac{1}{W_t} \sum_{j=1}^{m} \delta\left(s_t^{(j)} = a_i\right) w_t^{(j)}, \qquad i = 1, \ldots\ldots, |A| \tag{3.45}$$

where $W_t = \sum_{j=1}^{m} w_t^{(j)}$

$\delta(.)$ is dirac-delta function defined as

$$\delta\left(s_t^{(j)} = a_i\right) = \begin{cases} 1, & \text{if} \quad s_t^{(j)} = a_i \\ 0, & \text{if} \quad s_t^{(j)} \neq a_i \end{cases} \tag{3.46}$$

The decision on the symbol $s_t$ is obtained as

$$\hat{s}_t = \arg\max_{a_i \in A} p\left(s_t = a_i / \mathbf{Y}_t\right)$$

$$\cong \arg\max_{a_i \in A} \sum_{j=1}^{m} \delta\left(s_t^{(j)} = a_i\right) w_t^{(j)} \tag{3.47}$$

The estimated symbol $\hat{s}_t$ may have a phase ambiguity since M-ary phase shift keying is used. For instance, binary phase shift keying (BPSK) signals, $s_t \in \{-1,+1\}$. It can be easily seen that from (3.1) that if both the symbol sequence $\{s_t\}$ and the channel value sequence $\{\alpha_t\}$ are phase-shifted by $\pi$, no change is incurred on the observed signal $\{y_t\}$.Alternatively, in the state space model (3.12) & (3.13) a phase shift of $\pi$ on both the symbol sequence $\{s_t\}$ and the state sequence $\{\mathbf{x}_t\}$yields the same model for the observations. Hence such a phase ambiguity necessitates the differential encoding and decoding.

The particle filter algorithm for generating the sequential Monte Carlo samples of transmitted symbols $\left\{\mathbf{S}_t^{(j)}\right\}_{j=1}^{m}$ with corresponding importance weights $\left\{w_t^{(j)}\right\}_{j=1}^{m}$ and Kalman filter update $k_t^{(j)} = \left(\mathbf{x}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\right)$are given in algorithm 3.1.[16].

***Algorithm 3.1***

*1) Initialization*

Each Kalman filter is initialized as $k_0^{(j)} = \left(\mathbf{x}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)}\right)$, with $\mathbf{x}_0^{(j)} = \mathbf{0}$, $\boldsymbol{\Sigma}_0^{(j)} = \mathbf{I}$, $j = 1,.......m$. All the importance weights are initialized as $w_0^j = 1, j = 1,.......m$

so that there is no bias in decision making by initial weights. Since the data symbols are assumed to be independent, initial symbols are not needed.

Based on the state space model (3.12) & (3.13), the following steps are implemented at time $t$ to update each weighted sample. For $j = 1, ......m$

2) *Compute the one-step predictive update of each Kalman filter* $k_{t-1}^{(j)}$

From equations (3.34), (3.35) and (3.36), the predictive update of Kalman filter $k_{t-1}^{(j)}$ is given by

$$\mathbf{K}_t^{(j)} = \mathbf{F}\mathbf{\Sigma}_{t-1}^{(j)}\mathbf{F}^H + \mathbf{g}\mathbf{g}^H$$

$$R_t^{(j)} = \mathbf{h}^H \mathbf{K}_t^{(j)} \mathbf{h} + \sigma^2$$

$$\mu_t^{(j)} = \mathbf{h}^H \mathbf{F} \mathbf{x}_{t-1}^{(j)}$$

3) *Compute the trial sampling density*

For each $a_i \in A$, compute $\rho_{t,i}^{(j)}$ by using equation (3.44) as

$$\rho_{t,i}^{(j)} = p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) p(s_t = a_i).$$

$$p\left(y_t / s_t = a_i, \mathbf{S}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \sim N_c\left(a_i \mu_t^{(j)}, R_t^{(j)}\right)$$

4) *Impute the symbol* $s_t$

Draw $s_t^{(j)}$ from the Alphabet set $A$ with probability

$$p(s_t^{(j)} = a_i) \propto \rho_{t,i}^{(j)}, \quad a_i \in A \tag{3.48}$$

Append $s_t^{(j)}$ to $S_{t-1}^{(j)}$ and obtain $S_t^{(j)}$

5) *Compute the importance weight*

By using equation (3.31), the weight update is given by

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

6) *Compute the one-step filtering update of the Kalman filter* $k_{t-1}^{j}$

Based on the imputed symbol $s_t^{(j)}$ and the observation $y_t$, the Kalman filter update $k_t^{(j)} = \left(\mathbf{x}_t^{(j)}, \Sigma_t^{(j)}\right)$ is obtained by using equations (3.39) & (3.37) as

$$\mathbf{x}_t^{(j)} = \mathbf{F}\mathbf{x}_{t-1}^{(j)} + \frac{1}{R_t^{(j)}}\left(y_t - s_t^{(j)}\mu_t^{(j)}\right) s_t^{(j)*}\mathbf{K}_t^{(j)}\mathbf{h}$$

$$\Sigma_t^{(j)} = \mathbf{K}_t^{(j)} - \frac{1}{R_t^{(j)}}\mathbf{K}_t^{(j)}\mathbf{h}\mathbf{h}^H\mathbf{K}_t^{(j)}$$

At each time *t*, the only quantities that need to be stored are $\left\{k_t^j, w_t^j\right\}_{j=1}^{m}$ At each time *t*, the dominant computation involves the *m* one-step Kalman filter updates. Since *m* samplers operate independently and in parallel, the SMC detector is well suited for parallel implementations.

## 3.3 Resampling

The importance sampling weight $w_t^{(j)}$ measures the "quality" of the corresponding imputed signal sequence $\mathbf{S}_t^{(j)}$. A relatively small weight implies that the sample is drawn far from the main body of the posterior distribution and has small contribution in the final estimation. Such a sample is said to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient. This can be detected by observing a large *coefficient of variation* [16] in the importance weight. Suppose, $\left\{w_t^{(j)}\right\}_{j=1}^{m}$ is a sequence of the importance weights. Then the coefficient of variation, $v_t$ is defined as [16]

$$v_t^2 = \frac{\sum_{j=1}^{m}\left(w_t^{(j)} - \bar{w}_t\right)^2 / m}{\bar{w}_t^2} \tag{3.49}$$

$$v_t^2 = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{w_t^{(j)}}{\bar{w}_t} - 1\right)^2 \tag{3.50}$$

Where, $\bar{w}_t = \sum_{j=1}^{m} w_t^{(j)} \Big/ m$ (3.51)

A measure of the efficiency of an importance sampling scheme is the *effective sample size* $\bar{m}_t$, is defined as [16]

$$\bar{m}_t = \frac{m}{1+v_t^2} \tag{3.52}$$

In dynamic resampling, a resampling step is performed once the effective sample size is below a certain threshold, e.g, $\bar{m}_t \leq \frac{m}{10}$. Alternatively, resampling can be done at every fixed- length interval (say, every five steps).

In [16], Rong Chen *et al*, proposed a *residual resampling* strategy, which forms a new set of weighted samples $\left\{\left(\tilde{\mathbf{S}}_t^{(j)}, \tilde{k}_t^{(j)}, \tilde{w}_t^{(j)}\right)\right\}_{j=1}^{m}$ from original set $\left\{\left(\mathbf{S}_t^{(j)}, k_t^{(j)}, w_t^{(j)}\right)\right\}_{j=1}^{m}$ according to the algorithm 3.2 (assume that $\sum_{j=1}^{m} w_t^{(j)} = m$ after proper normalization).

***Algorithm 3.2***[16]

1) For $j = 1, \ldots\ldots\ldots, m$ retain $k_j = \left\lfloor w_t^{(j)} \right\rfloor$ copies of the sample $\left(\mathbf{S}_t^{(j)}, k_t^{(j)}\right)$. Denote $K_r = m - \sum_{j=1}^{m} k_j$.

2) Obtain $K_r$ i.i.d draws from the original sample set $\left\{\left(\mathbf{S}_t^{(j)}, k_t^{(j)}\right)\right\}_{j=1}^{m}$, with probabilities proportional to

$$\left(w_t^{(j)} - k_j\right), \qquad for \quad j = 1,......,m$$

3) Assign equal weight, i.e., $\tilde{w}_t^{(j)} = 1$, for each new sample.

**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 [16], 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 [16] and delayed-sample method [16].

**Delayed-Weight Method**

If the set $\left\{\left(\mathbf{S}_t^{(j)}, w_t^{(j)}\right)\right\}_{j=1}^m$ is properly weighted with respect to $p\left(\mathbf{S}_t/\mathbf{Y}_t\right)$, then by induction, the set $\left\{\left(\mathbf{S}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)}\right)\right\}_{j=1}^m$ is properly weighted with respect to $p\left(\mathbf{S}_{t+\delta}/\mathbf{Y}_{t+\delta}\right), \delta > 0$. Hence, by focussing on $\mathbf{S}_t$ at time $(t+\delta)$, the delayed estimate of the symbol can be obtained as [16]

$$p\left(s_t = a_i/\mathbf{Y}_{t+\delta}\right) \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^{m} \delta\left(s_t^{(j)} = a_i\right) w_{t+\delta}^{(j)}, \qquad i = 1,.......,|A|. \tag{3.53}$$

where $W_{t+\delta} = \sum_{j=1}^{m} w_{t+\delta}^{(j)}$ .Since the weights $\left\{w_{t+\delta}^{(j)}\right\}_{j=1}^m$ contain the information about the signals $\left(y_{t+1},.........,y_{t+\delta}\right)$, the estimate in equation (3.53) is usually more accurate. The delayed estimation method incurs no additional computational cost (i.e., CPU time), but it requires some extra memory for storing $\left\{\left(s_{t+1}^{(j)},.........,s_{t+\delta}^{(j)}\right)\right\}_{j=1}^m$ .For uncoded systems this simple delayed-weight method is quite effective for improving the detection performance over the concurrent method.

## 3.4 Particle Filtering for Blind Detection in MIMO Systems

There is an increasing demand for the design of multiple-input multiple output (MIMO) communication system for high data-rate wireless communications. An MIMO system employs multiple antennas at the transmitter and the receiver, and its capacity increases linearly with the minimum between the numbers of transmit and receive antennas [9,18]. When channels of MIMO systems are known, maximum likelihood sequence detector (MLSE) [18] is optimal, which searches for the data sequence that after convolution with the channel is closest in Euclidean distance to the received signal sequence but its complexity increases exponentially with the dimension of the parameter to be estimated. Zero-forcing (ZF) detector [18] and the minimum mean square error (MMSE) detector [18] require only linear complexity, but cannot achieve optimal performance. However, the channel dynamics cannot be known in advance and they will change from time to time. The channel state information (CSI) can be known by transmitting a pilot sequence or training sequence periodically [9], which result in wastage of power, bandwidth, reduces the communication throughput and creates overhead problems.

A novel sequential Monte Carlo blind receiver for MIMO systems is presented in [19]. The basic idea in [19] is to design a probabilistic approximation method for computation of the maximum a posteriori distribution (MAP) [19] via sequential Monte Carlo method (SMC). We have used the state space model approach for deriving the particle filtering algorithm for the blind detection in differentially encoded MIMO systems with the use of Kalman filtering algorithm in both flat fading and frequency-selective fading channels. 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 the MIMO system is described. Next, the state space model of MIMO system in both flat fading and frequency-selective fading channel is derived. The particle filter algorithm for differentially encoded MIMO system is derived and finally the simulation results of SISO and MIMO systems are presented.

### 3.4.1 Signal Model of MIMO System

Consider an MIMO system with $K$ transmit and $P$ receive antennas over fading channels with additive Gaussian noise as given in Fig 3.2[19].

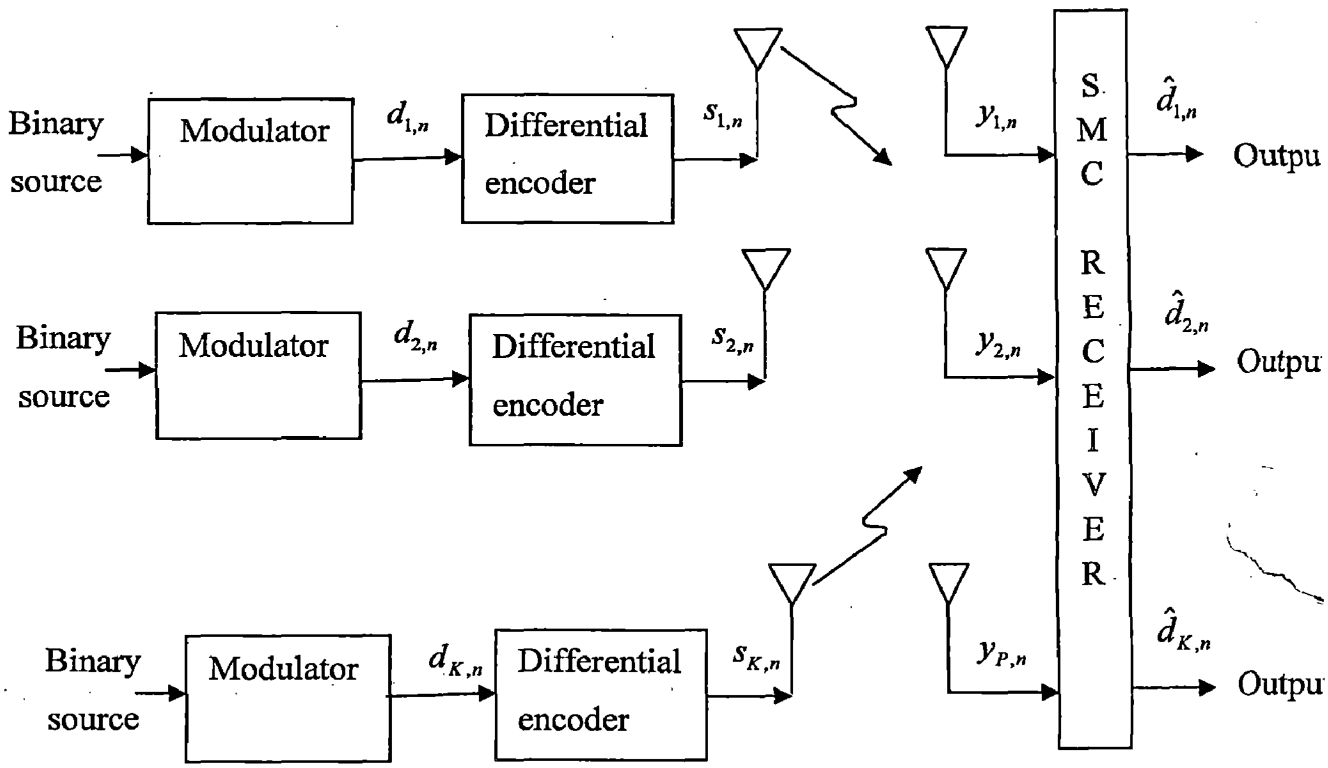



**Figure 3.2** An MIMO system with sequential Monte Carlo receiver

As the Fig. 3.2 shows, at each transmit antenna $k$, $k = 1,........,K$, the binary bits emitted from binary source are mapped into multi-phase signals $d_{k,n}$; $n = 1,........,N-1$ in the modulator, which take values from a finite alphabet set $A = \{a_1,...,a_{|A|}\}$, where $N$ is the block size. These signals $\{d_{k,n}\}_{n=1}^{N-1}$ are differentially encoded to resolve the phase ambiguity inherent to any blind receiver, and the output of differential encoder $s_{k,n}$ is given by [19]

$$\begin{aligned} s_{k,0} &= 1, \\ s_{k,n} &= s_{k,n-1} d_{k,n}, \qquad n = 1,....,N-1 \end{aligned} \tag{3.54}$$

The differentially encoded symbols are transmitted through the transmitting antennas and are assumed to be independent in time as well as in space.

Assume that the channel between $k^{th}$ transmit antenna and the $p^{th}$ receive antenna is subject to flat fading, then the received signal $y_{p,n}$ at the $p^{th}$ receive antenna, $p=1,......,P$ and at time $n$ is given by [19]

$$y_{p,n}=\sum_{k=1}^{K}h_{p,k,n}s_{k,n}+v_{p,n},\qquad p=1,......,P,\qquad n=0,....,N-1 \tag{3.55}$$

where $s_{k,n}$ is the transmitted symbol at the $k^{th}$ transmit antenna at time $n$

$h_{p,k,n}$ is the complex fading channel gain between the $k^{th}$ transmit antenna and the $p^{th}$ receive antenna

$v_{p,n}$ is i.i.d complex Gaussian noise $\left(v_{p,n}\sim N_c\left(0,\sigma^2\right)\right)$

By writing equation (3.55) in matrix form, we get

$$\begin{bmatrix} y_{1,n} \\ y_{2,n} \\ \vdots \\ y_{P,n} \end{bmatrix}=\begin{bmatrix} h_{1,1,n} & h_{1,2,n} & \cdots & h_{1,K,n} \\ h_{2,1,n} & h_{2,2,n} & \cdots & h_{2,K,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{P,1,n} & h_{P,2,n} & \cdots & h_{P,K,n} \end{bmatrix}\begin{bmatrix} s_{1,n} \\ s_{2,n} \\ \vdots \\ s_{K,n} \end{bmatrix}+\begin{bmatrix} v_{1,n} \\ v_{2,n} \\ \vdots \\ v_{P,n} \end{bmatrix} \tag{3.56}$$

Let,

$$\begin{aligned} \mathbf{h}_{1,n} &= \begin{bmatrix} h_{1,1,n} & h_{1,2,n} & \cdots & h_{1,K,n} \end{bmatrix}^T \\ \mathbf{h}_{2,n} &= \begin{bmatrix} h_{2,1,n} & h_{2,2,n} & \cdots & h_{2,K,n} \end{bmatrix}^T \\ &\vdots \\ \mathbf{h}_{P,n} &= \begin{bmatrix} h_{P,1,n} & h_{P,2,n} & \cdots & h_{P,K,n} \end{bmatrix}^T \end{aligned}$$

and $\quad \mathbf{s}_n=\begin{bmatrix} s_{1,n} & s_{2,n} & \cdots & s_{K,n} \end{bmatrix}$

Then, the equation (3.56) is modified as

$$\begin{bmatrix} y_{1,n} \\ y_{2,n} \\ \vdots \\ y_{P,n} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_n & 0 & \cdots & 0 \\ 0 & \mathbf{s}_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{s}_n \end{bmatrix} \begin{bmatrix} \mathbf{h}_{1,n} \\ \mathbf{h}_{2,n} \\ \vdots \\ \mathbf{h}_{P,n} \end{bmatrix} + \begin{bmatrix} v_{1,n} \\ v_{2,n} \\ \vdots \\ v_{P,n} \end{bmatrix} \tag{3.57}$$

Denoting $\boldsymbol{\psi}_n = \begin{bmatrix} \mathbf{s}_n & 0 & \cdots & 0 \\ 0 & \mathbf{s}_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{s}_n \end{bmatrix}$

$$\mathbf{h}_n = \begin{bmatrix} \mathbf{h}_{1,n} & \mathbf{h}_{2,n} & \cdots & \mathbf{h}_{P,n} \end{bmatrix}^T$$

$$\mathbf{v}_n = \begin{bmatrix} v_{1,n} & v_{2,n} & \cdots & v_{P,n} \end{bmatrix}^T$$

and $\mathbf{y}_n = \begin{bmatrix} y_{1,n} & y_{2,n} & \cdots & y_{P,n} \end{bmatrix}^T$

then, equation (3.57) is written as

$$\mathbf{y}_n = \boldsymbol{\psi}_n \mathbf{h}_n + \mathbf{v}_n \tag{3.58}$$

Assume that the fading coefficients $\{h_{p,k,n}\}$ remain fixed for a block of $N$ symbols, then

$$\mathbf{h}_{n+1} = \mathbf{h}_n \tag{3.59}$$

Now, the state space model of MIMO system in flat fading channel for a block of $N$ symbols using equations (3.58) & (3.59) is given by

$$\mathbf{h}_{n+1} = \mathbf{h}_n \tag{3.60}$$

$$\mathbf{y}_n = \boldsymbol{\psi}_n \mathbf{h}_n + \mathbf{v}_n \tag{3.61}$$

## 3.5 Particle Filtering Algorithm for MIMO System

Consider the state space model of MIMO system given by equations (3.60) & (3.61). Let $\mathbf{Y}_n \triangleq \{\mathbf{y}_0, \mathbf{y}_1, \dots\dots\dots, \mathbf{y}_n\}$, be received signals and $\mathbf{S}_n \triangleq \{\mathbf{s}_0, \mathbf{s}_1, \dots\dots\dots, \mathbf{s}_n\}$ be transmitted signals up to time $n$. Also denote $\mathbf{d}_n \triangleq \{d_{1,n}, d_{2,n}, \dots\dots\dots, d_{K,n}\}$, $\mathbf{D}_n \triangleq \{\mathbf{d}_1, \mathbf{d}_2, \dots\dots\dots, \mathbf{d}_n\}$.

***Statement of the Problem:*** *To estimate the a posteriori probabilities of the information symbols*

$$p(\mathbf{d}_n = \mathbf{a}_i / \mathbf{Y}_n), \qquad \mathbf{a}_i \in A^K; n = 1, \dots, N-1 \tag{3.62}$$

*based on the received signals* $\mathbf{Y}_n$ *up to time n and the a priori symbol probabilities of* $\mathbf{D}_n$ *without the knowledge of channel response* $\mathbf{h}_n$.

Consider $M$-ary phase-shift keying (MPSK) signals are transmitted i.e.,

$$a_i = exp\left(j\frac{2\pi i}{|A|}\right), \qquad for \;\; i = 0, \dots\dots\dots, |A|-1 \tag{3.63}$$

where $j = \sqrt{-1}$. Assume that the transmitted symbols are independent i.e.,

$$p(\mathbf{s}_n = \mathbf{a}_i / \mathbf{S}_{n-1}) = p(\mathbf{s}_n = \mathbf{a}_i), \qquad \mathbf{a}_i \in A^K \tag{3.64}$$

Now, using equation (5.1) the probability $p(\mathbf{s}_n = \mathbf{a}_i)$ is given by

$$p(\mathbf{s}_n = \mathbf{a}_i) = p(\mathbf{d}_n = \mathbf{a}_i \circ \mathbf{s}_{n-1}^*) \tag{3.65}$$

where $\circ$ denotes element-wise product. When no prior information about the symbols is available, the symbols are assumed to take each possible value in the Alphabet set $A^K$ with equal probability i.e.,

$$p(\mathbf{d}_n = \mathbf{a}_i \circ \mathbf{s}_{n-1}^*) = \frac{1}{|A^K|} \quad \text{for} \;\; i = 1, \dots\dots\dots |A^K| \tag{3.66}$$

Let $\mathbf{s}_n^{(j)} \triangleq \{s_{1,n}^{(j)}, s_{2,n}^{(j)}, \dots\dots\dots, s_{K,n}^{(j)}\}, j = 1, 2, \dots\dots, m$ be a sample drawn at time $n$ and denote $\mathbf{S}_n^{(j)} \triangleq \{\mathbf{s}_0^{(j)}, \mathbf{s}_1^{(j)}, \dots\dots\dots, \mathbf{s}_n^{(j)}\}$. In order to implement the particle filter, a set of

Monte Carlo samples of the transmitted symbols $\left\{\mathbf{S}_n^{(j)}\right\}_{j=1}^{m}$ with its corresponding importance weights $\left\{w_n^{(j)}\right\}_{j=1}^{m}$ which are properly weighted with respect to the distribution $p(\mathbf{S}_n/\mathbf{Y}_n)$ are needed. From SIS technique discussed in section 2.4.1, the Monte Carlo samples are easily generated from a trial sampling density. If the choice of trial sampling density is taken as the optimal sampling density then by equation (2.39), we get

$$q\left(\mathbf{s}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_n\right)=p\left(\mathbf{s}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_n\right) \tag{3.67}$$

For this choice of sampling density, the weights are updated according to equation (2.40) as

$$w_n^{(j)} \propto w_{n-1}^{(j)} p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_{n-1}\right) \tag{3.68}$$

$$w_n^{(j)} \propto w_{n-1}^{(j)} \sum_{\mathbf{a}_i\in A^K} p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)p\left(\mathbf{s}_n=\mathbf{a}_i/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_{n-1}\right)$$

$$w_n^{(j)} \propto w_{n-1}^{(j)} \sum_{\mathbf{a}_i\in A^K} p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)p\left(\mathbf{s}_n=\mathbf{a}_i\right)$$

$$w_n^{(j)} \propto w_{n-1}^{(j)} \sum_{\mathbf{a}_i\in A^K} p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)p\left(\mathbf{d}_n=\mathbf{a}_i\circ\mathbf{s}_{n-1}^{(j)*}\right) \tag{3.69}$$

Denote $\alpha_{n,i}^{(j)}=p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)p\left(\mathbf{d}_n=\mathbf{a}_i\circ\mathbf{s}_{n-1}^{(j)*}\right)$, then weights are given by

$$w_n^{(j)} \propto w_{n-1}^{(j)} \sum_{\mathbf{a}_i\in A^K} \alpha_{n,i}^{(j)} \tag{3.70}$$

Now, the term sampling density $p\left(\mathbf{s}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_n\right)$ is modified as

$$p\left(\mathbf{s}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{Y}_n\right)=p\left(\mathbf{s}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{y}_n,\mathbf{Y}_{n-1}\right)$$

$$= \frac{p\left(\mathbf{s}_n, \mathbf{S}_{n-1}^{(j)}, \mathbf{y}_n, \mathbf{Y}_{n-1}\right)}{p\left(\mathbf{S}_{n-1}^{(j)}, \mathbf{y}_n, \mathbf{Y}_{n-1}\right)}$$

$$= \frac{p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n, \mathbf{Y}_{n-1}\right) p\left(\mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n, \mathbf{Y}_{n-1}\right)}{p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_{n-1}\right) p\left(\mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_{n-1}\right)}$$

$$= \frac{p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n, \mathbf{Y}_{n-1}\right) p\left(\mathbf{s}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_{n-1}\right)}{p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_{n-1}\right)}$$

$$\propto p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) p\left(\mathbf{s}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_{n-1}\right)$$

$$\propto p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) p\left(\mathbf{s}_n = \mathbf{a}_i\right)$$

$$\propto p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) p\left(\mathbf{d}_n = \mathbf{a}_i \circ \mathbf{s}_{n-1}^{(j)*}\right)$$

$$p\left(\mathbf{s}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{Y}_n\right) = \alpha_{n,i}^{(j)} \qquad (3.71)$$

From the state space model (3.60) & (3.61), the density $p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right)$ is Gaussian and its mean and variance is calculated using the Kalman filtering algorithm.

$$p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) \sim N_c\left(mean, variance\right) \qquad (3.72)$$

The state space model of MIMO system defined by equations (3.60) & (3.61) is reproduced here.

$$\mathbf{h}_{n+1} = \mathbf{h}_n$$

$$\mathbf{y}_n = \psi_n \mathbf{h}_n + \mathbf{v}_n$$

The Kalman filtering algorithm for the above state space model when $\mathbf{s}_n$ is known, is given as follows:

The innovation term from equation (2.17) is given by

$$\boldsymbol{\eta}_n = \mathbf{y}_n - \boldsymbol{\psi}_n \hat{\mathbf{h}}_{n-1} \tag{3.73}$$

The correlation matrix of the innovation process by using equation (2.19) is given by

$$\mathbf{R}_n = \boldsymbol{\psi}_n \mathbf{K}_n \boldsymbol{\psi}_n^H + \sigma^2 \mathbf{I}_P \tag{3.74}$$

The Kalman gain by using equation (2.20) is given by

$$\mathbf{G}_n = \mathbf{K}_n \boldsymbol{\psi}_n^H \mathbf{R}_n^{-1} \tag{3.75}$$

Filtered state error correlation matrix by using equation (2.18) is given by

$$\boldsymbol{\Sigma}_n = \mathbf{K}_n - \mathbf{G}_n \boldsymbol{\psi}_n \mathbf{K}_n$$

$$\boldsymbol{\Sigma}_n = \mathbf{K}_n - \mathbf{K}_n \boldsymbol{\psi}_n^H \mathbf{R}_n^{-1} \boldsymbol{\psi}_n \mathbf{K}_n \tag{3.76}$$

Predicted state error correlation matrix by using equation (2.16) is given by

$$\mathbf{K}_n = \boldsymbol{\Sigma}_{n-1} \tag{3.77}$$

Now, substitute equation (3.77) in equations (3.74), (3.75) and (3.76) then

$$\mathbf{R}_n = \boldsymbol{\psi}_n \boldsymbol{\Sigma}_{n-1} \boldsymbol{\psi}_n^H + \sigma^2 \mathbf{I}_P \tag{3.78}$$

$$\mathbf{G}_n = \boldsymbol{\Sigma}_{n-1} \boldsymbol{\psi}_n^H \mathbf{R}_n^{-1} \tag{3.79}$$

$$\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_{n-1} - \boldsymbol{\Sigma}_{n-1} \boldsymbol{\psi}_n^H \mathbf{R}_n^{-1} \boldsymbol{\psi}_n \boldsymbol{\Sigma}_{n-1} \tag{3.80}$$

Estimated state vector by using equation (2.17) is given by

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \mathbf{G}_n \boldsymbol{\eta}_n$$

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \boldsymbol{\Sigma}_{n-1} \boldsymbol{\psi}_n^H \mathbf{R}_n^{-1} \left( \mathbf{y}_n - \boldsymbol{\psi}_n \hat{\mathbf{h}}_{n-1} \right) \tag{3.81}$$

The *mean* of density $p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right)$ can be computed using the innovation equation (3.73) as [19]

$$mean = E\left\{\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right\} \tag{3.82}$$

$$= \boldsymbol{\psi}_n \mathbf{h}_{n-1}^{(j)}\Big|_{\mathbf{s}_n = \mathbf{a}_i}$$

$$= \boldsymbol{\psi}(\mathbf{a}_i)\mathbf{h}_{n-1}^{(j)}$$

Now, denote $\boldsymbol{\Xi}_i = \boldsymbol{\psi}(\mathbf{a}_i)$, then

$$mean = \boldsymbol{\Xi}_i \mathbf{h}_{n-1}^{(j)} \tag{3.83}$$

Let $\boldsymbol{\mu}_{n,i}^{(j)} = \boldsymbol{\Xi}_i \mathbf{h}_{n-1}^{(j)}$ then *mean* is given by

$$mean = \boldsymbol{\mu}_{n,i}^{(j)} \tag{3.84}$$

The *variance* of density $p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right)$ can be computed by using the correlation matrix of the innovation process equation (3.78) as [19]

$$variance = \mathrm{Var}\left\{\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right\} \tag{3.85}$$

$$= \boldsymbol{\psi}_n \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\psi}_n^H + \sigma^2 \mathbf{I}_P\Big|_{\mathbf{s}_n = \mathbf{a}_i}$$

$$= \boldsymbol{\psi}(\mathbf{a}_i)\boldsymbol{\Sigma}_{n-1}^{(j)}\boldsymbol{\psi}^H(\mathbf{a}_i) + \sigma^2 \mathbf{I}_P$$

$$variance = \boldsymbol{\Xi}_i \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\Xi}_i^H + \sigma^2 \mathbf{I}_P \tag{3.86}$$

Let $\mathbf{R}_{n,i}^{(j)} = \boldsymbol{\Xi}_i \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\Xi}_i^H + \sigma^2 \mathbf{I}_P$ then *variance* is given by

$$variance = \mathbf{R}_{n,i}^{(j)} \tag{3.87}$$

Now, the probability density $p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)$ using equations (3.84) & (3.87) is given by

$$p\left(\mathbf{y}_n/\mathbf{S}_{n-1}^{(j)},\mathbf{s}_n=\mathbf{a}_i,\mathbf{Y}_{n-1}\right)\sim N_c\left(\boldsymbol{\mu}_{n,i}^{(j)},\mathbf{R}_{n,i}^{(j)}\right) \tag{3.88}$$

For each $\mathbf{a}_i\in A^K$, the *a posteriori* symbol probability $p(\mathbf{d}_n=\mathbf{a}_i/\mathbf{Y}_n)$ as in equation (2.36) can be estimated as [19]

$$\begin{aligned} p(\mathbf{d}_n=\mathbf{a}_i/\mathbf{Y}_n) &= p\left(\mathbf{s}_n\circ\mathbf{s}_{n-1}^*=\mathbf{a}_i/\mathbf{Y}_n\right) \\ &= E\left\{\delta\left(\mathbf{s}_n\circ\mathbf{s}_{n-1}^*=\mathbf{a}_i\right)/\mathbf{Y}_n\right\} \\ &\cong \frac{1}{W_n}\sum_{j=1}^{m}\delta\left(\mathbf{s}_n^{(j)}\circ\mathbf{s}_{n-1}^{(j)*}=\mathbf{a}_i\right)w_n^{(j)} \end{aligned} \tag{3.89}$$

Where $W_n=\sum_{j=1}^{m}w_n^{(j)}$

$\delta(.)$ is dirac-delta function defined as

$$\delta\left(\mathbf{s}_n^{(j)}\circ\mathbf{s}_{n-1}^{(j)*}=\mathbf{a}_i\right)=\begin{cases}1, & \text{if } \mathbf{s}_n^{(j)}\circ\mathbf{s}_{n-1}^{(j)*}=\mathbf{a}_i\\ 0, & \text{if } \mathbf{s}_n^{(j)}\circ\mathbf{s}_{n-1}^{(j)*}\neq\mathbf{a}_i\end{cases} \tag{3.90}$$

The information symbol $\mathbf{d}_n$ is estimated by maximizing the a posterior density $p(\mathbf{d}_n=\mathbf{a}_i/\mathbf{Y}_n)$, where $\mathbf{a}_i\in A^K$. The particle filter algorithm for generating the sequential Monte Carlo samples of the transmitted symbols $\left\{\mathbf{S}_n^{(j)}\right\}_{j=1}^{m}$ with corresponding weights $\left\{w_n^{(j)}\right\}_{j=1}^{m}$, which are properly weighted with respect to the distribution $p(\mathbf{S}_n/\mathbf{Y}_n)$ and Kalman filter update $k_n^{(j)}=\left(\mathbf{h}_n^{(j)},\boldsymbol{\Sigma}_n^{(j)}\right)$ is given in algorithm 5.1[19].

*Algorithm 3.3*

1) *Initialization*

Each Kalman filter is initialized as $k_{-1}^{(j)} = \left(\mathbf{h}_{-1}^{(j)}, \boldsymbol{\Sigma}_{-1}^{(j)}\right)$, with $\boldsymbol{\Sigma}_{-1}^{(j)} = 1000\mathbf{I}_{KP}$ and $\mathbf{h}_{-1}^{(j)} \sim N_c\left(\mathbf{0}, \boldsymbol{\Sigma}_{-1}\right)$ for $j = 1, .......m$. All the importance weights are initialized as $w_{-1}^{j} = 1, j = 1, .......m$ so that there is no bias in decision making by initial weights.

Based on the state space model (3.60) & (3.61), the following steps are implemented at $n^{th}$ recursion $(n = 0, ......., N-1)$ to update each weighted sample. For $j = 1, .......m$

2) *For each* $\mathbf{a}_i \in A^K$, *compute the following quantities*

From equations (3.84) and (3.87), the mean and variance of the trial sampling density is calculated.

$$\boldsymbol{\mu}_{n,i}^{(j)} = \boldsymbol{\Xi}_i \mathbf{h}_{n-1}^{(j)}$$

$$\mathbf{R}_{n,i}^{(j)} = \boldsymbol{\Xi}_i \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\Xi}_i^H + \sigma^2 \mathbf{I}_P$$

Where $\boldsymbol{\Xi}_i = \psi(\mathbf{a}_i)$.

3) *Compute the trial sampling density*

For each $\mathbf{a}_i \in A^K$, compute $\alpha_{n,i}^{(j)}$ by using equation (3.88) as

$$\alpha_{n,i}^{(j)} = p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) p\left(\mathbf{d}_n = \mathbf{a}_i \circ \mathbf{s}_{n-1}^{(j)*}\right).$$

$$p\left(\mathbf{y}_n / \mathbf{S}_{n-1}^{(j)}, \mathbf{s}_n = \mathbf{a}_i, \mathbf{Y}_{n-1}\right) \sim N_c\left(\boldsymbol{\mu}_{n,i}^{(j)}, \mathbf{R}_{n,i}^{(j)}\right)$$

4) *Impute the symbol* $\mathbf{s}_n$

Draw a sample $\mathbf{s}_n^{(j)}$ from the set $A^K$ with probability

$$p(\mathbf{s}_n^{(j)} = \mathbf{a}_i) \propto \alpha_{n,i}^{(j)}, \quad \mathbf{a}_i \in A^K \tag{3.91}$$

Append $\mathbf{s}_n^{(j)}$ to $\mathbf{S}_{n-1}^{(j)}$ and obtain $\mathbf{S}_n^{(j)}$

5) *Compute the importance weight*

By using equation (5.24), the weight update is given by

$$w_n^{(j)} \propto w_{n-1}^{(j)} \sum_{\mathbf{a}_i \in A^K} \alpha_{n,i}^{(j)}$$

6) *Update the a posteriori mean and covariance of channel*

If the imputed symbol $\mathbf{s}_n^{(j)} = \mathbf{a}_i$ in step 4, then set $\boldsymbol{\mu}_n^{(j)} = \boldsymbol{\mu}_{n,i}^{(j)}$, $\mathbf{R}_n^{(j)} = \mathbf{R}_{n,i}^{(j)}$ and update the a posteriori mean and covariance of channel by using equations (3.81) & (3.80) as

$$\mathbf{h}_n^{(j)} = \mathbf{h}_{n-1}^{(j)} + \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\Xi}_n^{(j)H} \left(\mathbf{R}_n^{(j)}\right)^{-1} \left(\mathbf{y}_n - \boldsymbol{\mu}_n^{(j)}\right) \tag{3.92}$$

$$\boldsymbol{\Sigma}_n^{(j)} = \boldsymbol{\Sigma}_{n-1}^{(j)} - \boldsymbol{\Sigma}_{n-1}^{(j)} \boldsymbol{\Xi}_n^{(j)H} \left(\mathbf{R}_n^{(j)}\right)^{-1} \boldsymbol{\Xi}_n^{(j)} \boldsymbol{\Sigma}_{n-1}^{(j)} \tag{3.93}$$

At each time $n$, the dominant computation in this particle filtering algorithm involves the $m \times A^K$ one-step Kalman filter updates for $\left(\mathbf{h}_n^{(j)}, \boldsymbol{\Sigma}_n^{(j)}\right)$. Since the $m$ samplers operate independently and in parallel, the SMC detector is well suited for parallel implementations.

The residual resampling algorithm, which forms a new set of weighted samples $\left\{\left(\tilde{\mathbf{S}}_n^{(j)}, \tilde{k}_n^{(j)}, \tilde{w}_n^{(j)}\right)\right\}_{j=1}^{m}$ from original set, $\left\{\left(\mathbf{S}_n^{(j)}, k_n^{(j)}, w_n^{(j)}\right)\right\}_{j=1}^{m}$ can be generated as in algorithm 3.2.

## 3.6 Simulation Results

### 3.6.1 SISO System

For the simulation of blind detection in SISO systems, the following models namely ARMA (3,3) [19] and AR(2) [5] processes are used for the fading coefficients

$$\alpha_t - 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3}$$

$$= 10^{-2}\left(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} + 0.89409u_{t-3}\right) \qquad (3.94)$$

Where $u_t \sim N(0,1)$

$$\alpha_t - 0.10\alpha_{t-1} - 0.80\alpha_{t-2} = u_t \quad \text{and} \quad u_t \sim N(0, 0.27) \qquad (3.95)$$

For the simulation of blind detection in SISO system using particle filtering algorithm in MATLAB environment, the following parameters are used.

- Number of particles m= 50
- Modulation scheme: BPSK
- SNR=$Var\{\alpha_t\}/Var\{n_t\}$
- Number of Monte Carlo simulations $N_m$=100
- Number of transmitted symbols=100000
- For delayed estimation, delay=2 is considered.
- For the case of non-Gaussian noise $\varepsilon = 0.1$ and $k = 10$.

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

1) Obtain the randomly generated BPSK signals and differentially encode them before transmission.
2) Generate the true states using equation (3.12) and observations using equation (3.13).

3) Generate sequential Monte Carlo samples of transmitted symbols $\left\{\mathbf{s}_t^{(j)}\right\}_{j=1}^{m}$ with corresponding importance weights $\left\{w_t^{(j)}\right\}_{j=1}^{m}$ at time $t$ by using the algorithm 3.1.

4) Calculate the effective sample size $\bar{m}_t$ using equation (3.52).

5) If $\bar{m}_t \leq \dfrac{m}{10}$ then do resampling by using algorithm (3.2) else go back to step 3.

6) For each $a_i \in A$ , the a posteriori symbol probability $p\left(s_t = a_i / \mathbf{Y}_t\right)$ is calculated using the equation (3.45) after differentially decoding is performed.

7) The symbol is decoded using the equation (3.47) 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. 3.3 shows BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as ARMA process and 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. 3.4 shows BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as ARMA process and 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. 3.5 shows BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as AR process and 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. 3.6 shows BER performance of particle filtering algorithm in SISO systems 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.

### 3.6.2 MIMO System

For the simulation of blind detection in MIMO systems, fading coefficients are generated from the circularly symmetric complex Gaussian distribution. All fading coefficients are assumed to be uncorrelated and are normalized such that total energy is equal to unity.

For the simulation of blind detection in MIMO system using particle filtering algorithm in MATLAB environment, the following parameters are used.

- Number of particles m= 50
- Modulation scheme: BPSK
- Number of transmitting antennas $K$=2
- Number of receiving antennas $P$=2
- Number of transmitted symbols=100000
- Number of Monte Carlo simulations $N_m$=100

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

1) Generate BPSK signals from alphabet set $A = \{-1, 1\}$ randomly and differentially encode them using equation (3.54) before transmitting from each antenna.
2) Generate the fading coefficients and observations according to state space model of MIMO system given by equations (3.60) & (3.61).
3) Generate the sequential Monte Carlo samples of transmitted symbols $\left\{\mathbf{S}_n^{(j)}\right\}_{j=1}^{m}$ with corresponding importance weights $\left\{w_n^{(j)}\right\}_{j=1}^{m}$ at time $n$ by using the algorithm 3.3.
4) Calculate the effective sample size $\bar{m}_n$ using equation (3.52).
5) If $\bar{m}_n \le \frac{m}{10}$ then do resampling by using algorithm 3.2 else go back to step 3.
6) For each $\mathbf{a}_i \in A^K$, the a posteriori symbol probability $p(\mathbf{d}_n = \mathbf{a}_i / \mathbf{Y}_n)$ is calculated using the equation (3.89) .
7) The symbol is decoded by maximization of a posterior probability $p(\mathbf{d}_n = \mathbf{a}_i / \mathbf{Y}_n)$ and 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.

Fig. 3.7 shows BER performance of the particle filtering in blind detection of flat-fading MIMO system and additive Gaussian noise. Besides this, the performance of MLSE receiver with perfect channel state information, which serves as a lower bound on the achievable performance for any blind receiver is also given. As SNR is varied from 0dB-15dB, the BER of particle filtering method decreases from 0.3579 to 0.0057 while known channel bound decreases from 0.1930 to 0.0018. It may be seen that there is a close similarity between the known channel bound and the particle filtering method.

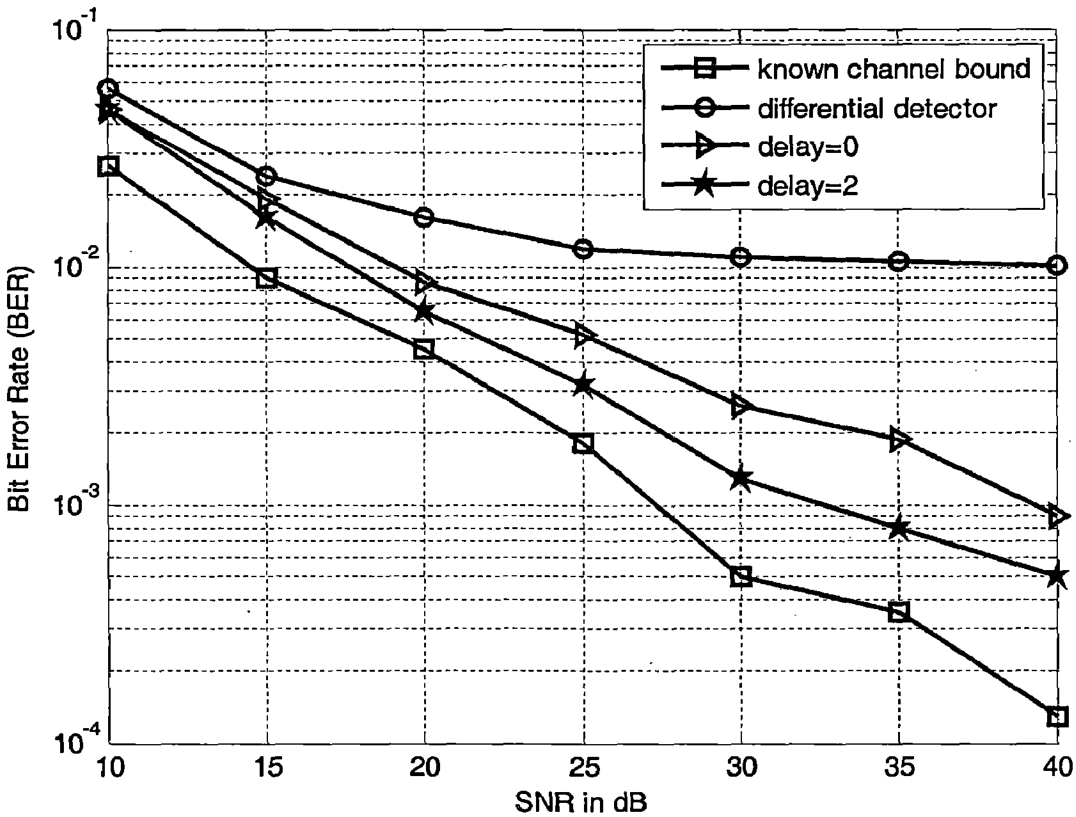



**Figure 3.3** BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as ARMA process and additive Gaussian noise.

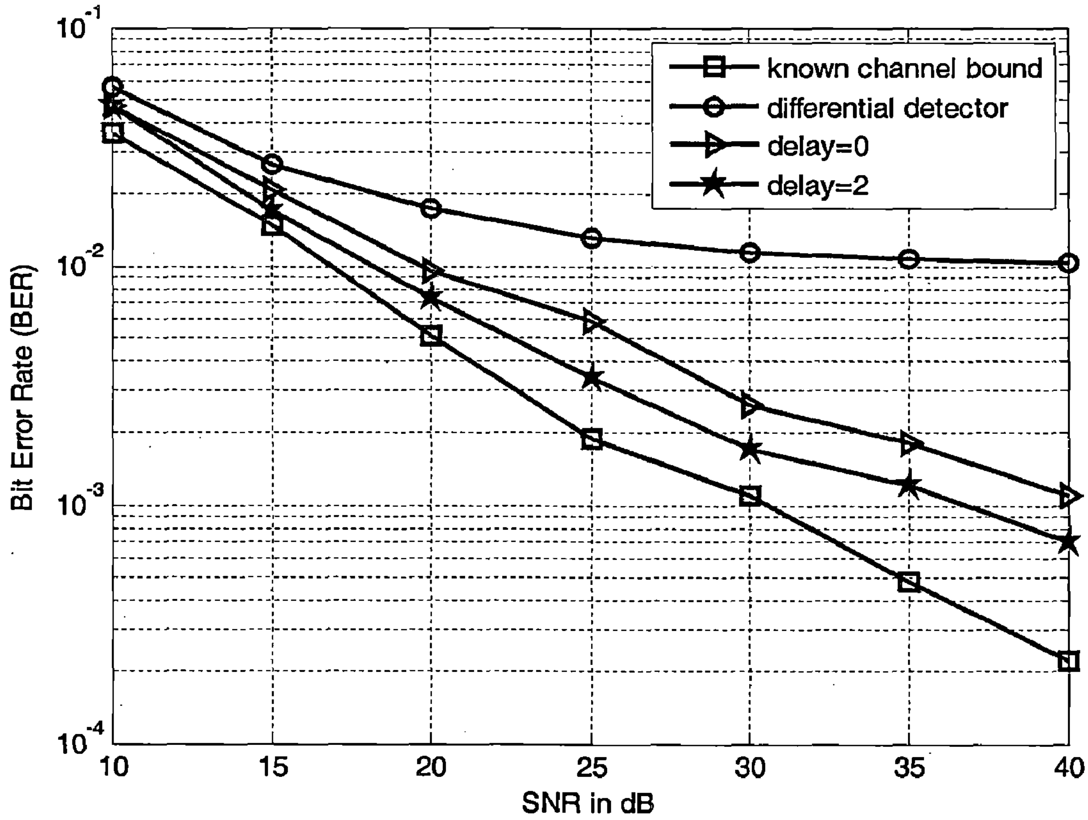



**Figure 3.4** BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as ARMA process and additive non-Gaussian noise.

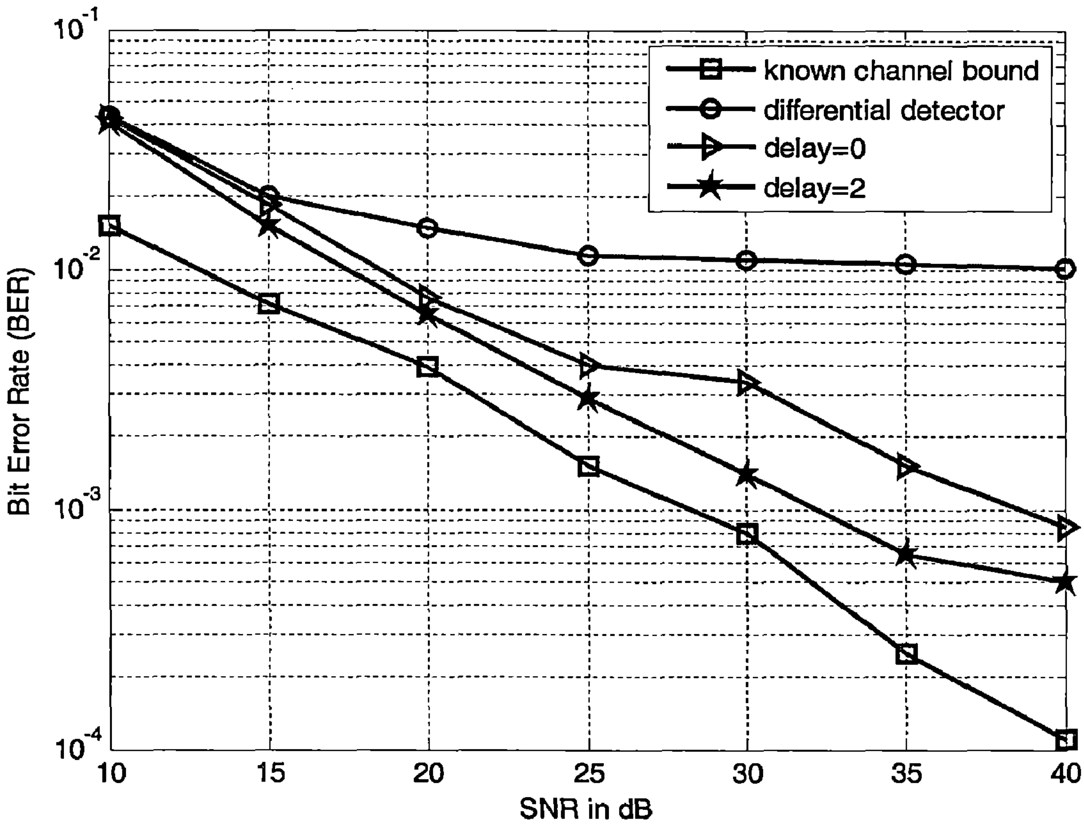



**Figure 3.5** BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as AR process and additive Gaussian noise.

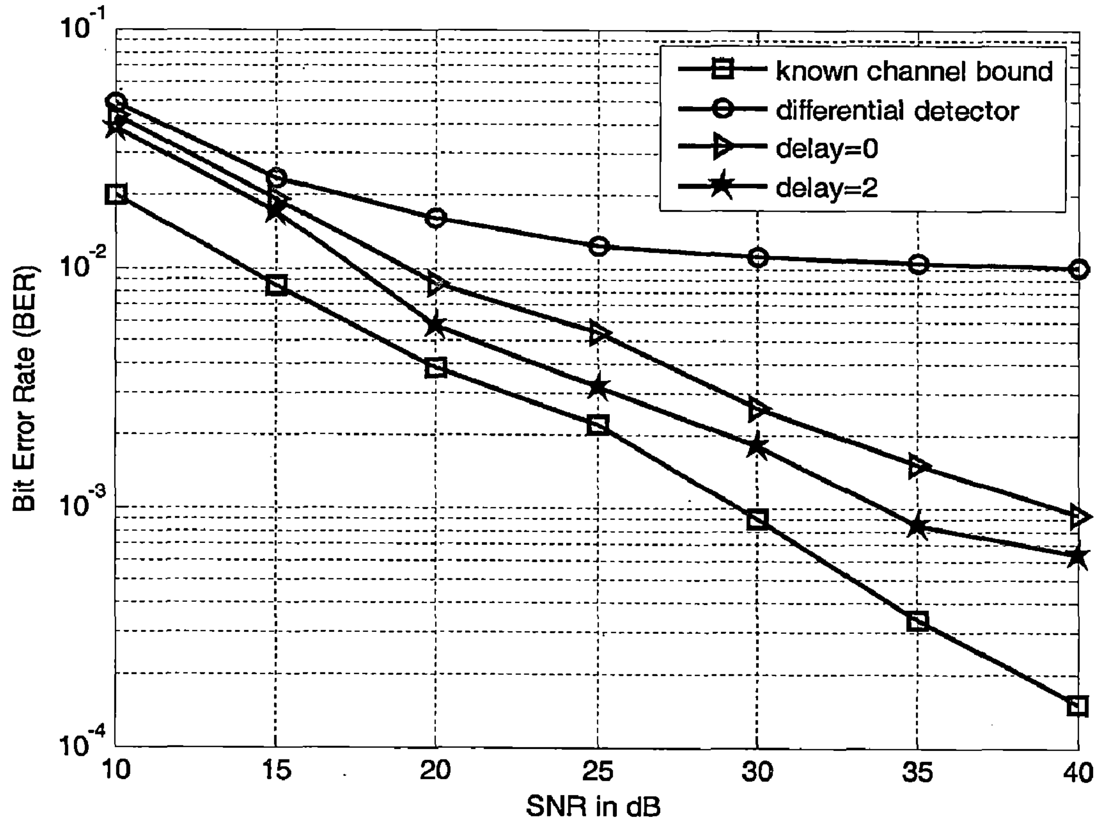



**Figure 3.6** BER performance of particle filtering algorithm in SISO systems with fading coefficients modeled as AR process and additive non-Gaussian noise.

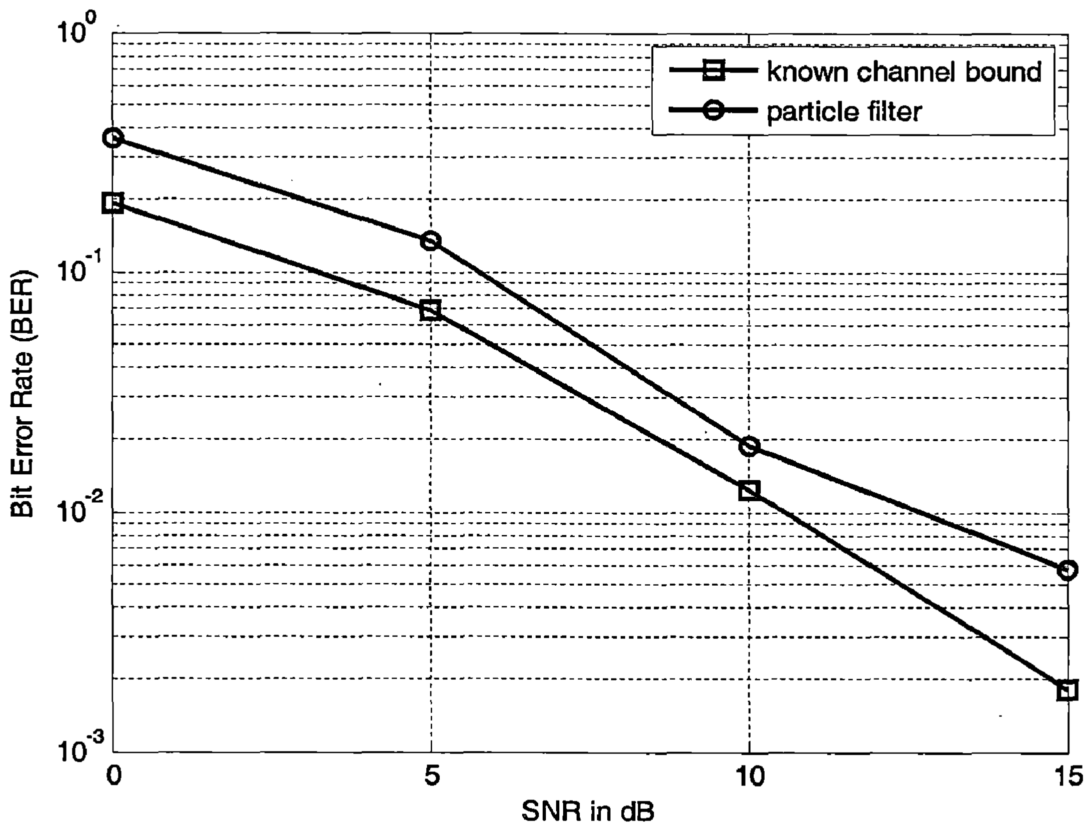



**Figure 3.7** The BER performance of the particle filtering in blind detection of flat-fading MIMO system. There are two transmitting antennas and two receiving antennas.

# Chapter 4

# PARTICLE FILTERING FOR BLIND DETECTION IN OFDM SYSTEMS

Orthogonal frequency division multiplexing (OFDM) [20] is one of the most promising techniques for achieving high speed wireless data communication. OFDM is a multicarrier transmission technique, which divides the single wideband channel into a number of narrowband channels called sub-channels; each subcarrier in each sub-channel is being modulated by a low rate data stream and sub-carriers are transmitted in parallel over the channel. The increased symbol duration reduces the impact of ISI. Thus OFDM is one of the most effective techniques for combating multipath delay spread over mobile wireless channels [20]. For OFDM systems, efficient and accurate channel estimation is necessary to coherently demodulate the received data. Channel estimation methods for OFDM systems based on the use of training signals are presented in [34,35]. However, transmissions of pilots require bandwidth and it significantly reduces the overall system capacity.

A sequential Monte Carlo blind receiver for OFDM systems in frequency-selective fading channels is presented in [22]. In this SMC detector, Bayesian inference of unknown data symbols in the presence of an unknown multipath fading channel is made only from the observations over one OFDM symbol duration based on techniques of importance sampling and resampling. We have used the state space model approach for deriving the particle filtering algorithm for the blind detection in differentially encoded OFDM systems with the use of Kalman filtering algorithm. 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, the OFDM system is described first. Next, the state space model of OFDM system for one symbol duration is presented. The particle filter algorithm for differentially encoded OFDM system in frequency-selective fading channel is derived. The residual resampling algorithm with fixed resampling interval and the delayed estimation approach are also discussed. Finally simulation results are presented.

## 4.1 Signal Model of OFDM System

Consider an OFDM system with $N$ subcarriers signaling through a frequency-selective fading channel as given in Fig4.1 [21,22].

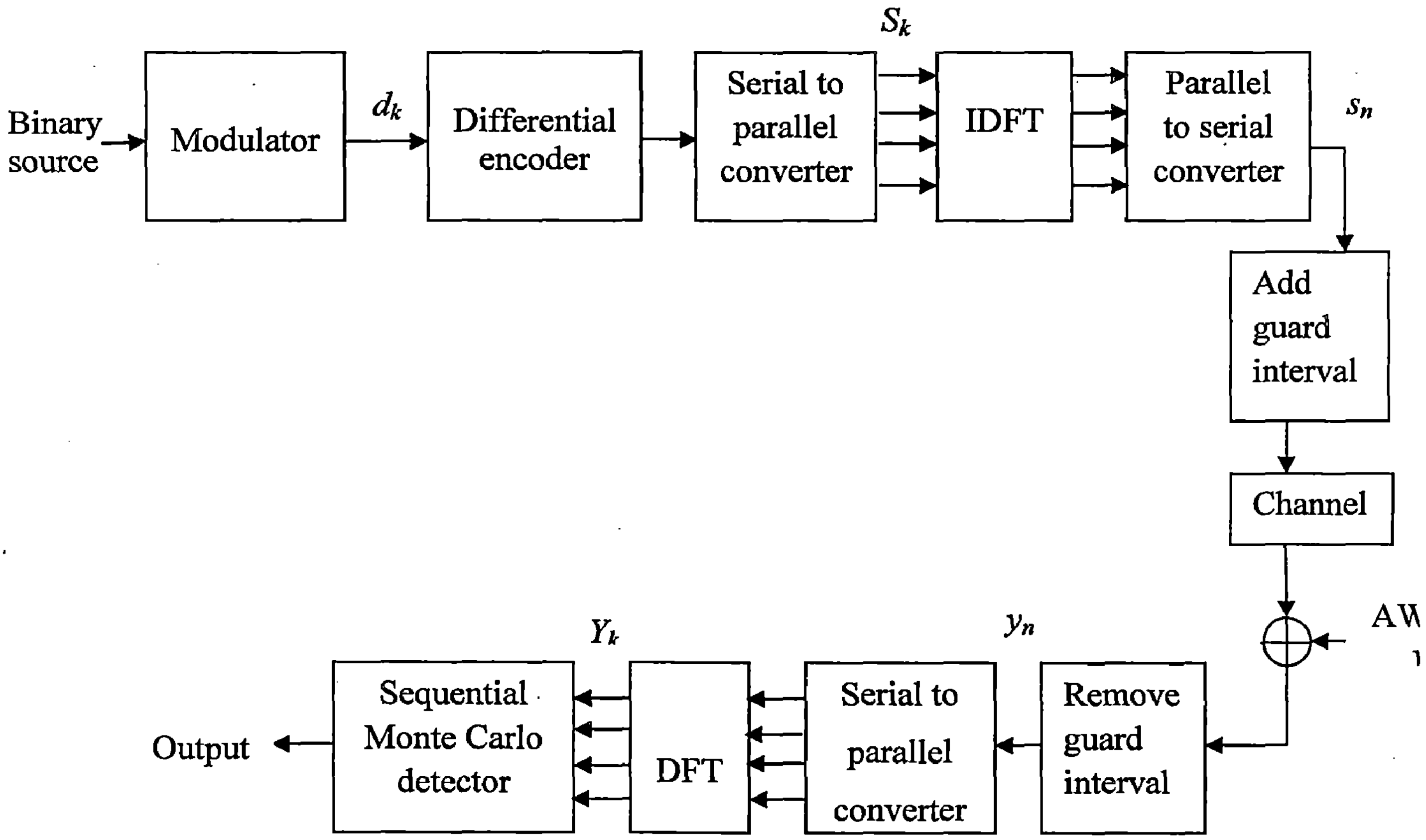



**Figure 4.1** Baseband model of OFDM system with sequential Monte Carlo detector

As shown in Fig.4.1, the binary digits emitted from source are mapped into multi-phase signals $d_k$ in the modulator, which takes values from a finite alphabet set $A = \{a_1, ..., a_{|A|}\}$. These signals $\{d_k\}_{k=1}^{N-1}$ are differentially encoded to resolve the phase ambiguity inherent to any blind receiver, and the output of differential encoder $S_k$ is given by [22]

$$S_0 = 1,$$
$$S_k = S_{k-1} d_k, \qquad k = 1, ...., N-1 \tag{4.1}$$

The differentially encoded symbols $\{S_k\}_{k=0}^{N-1}$ are applied to the Inverse Discrete Fourier Transform (IDFT) block resulting in $\{s_n\}_{n=0}^{N-1}$, which are given by

$$s_n = \text{IDFT}\{S_k\}$$

$$s_n = \frac{1}{N}\sum_{k=0}^{N-1} S_k e^{j2\pi kn/N}, \qquad n = 0,....,N-1 \tag{4.2}$$

When the number of sub-carriers $N$ increases, the OFDM symbol duration becomes large compared to the maximum multipath spread $\tau_{\mathrm{m}}$ of the channel, and the amount of intersymbol interference (ISI) reduces. However, to avoid the effects of ISI and to maintain the orthogonality between the sub-carriers, a guard interval is inserted between adjacent OFDM frames [21]. After parallel-to-serial conversion and insertion of guard interval, the signals are then transmitted through a frequency-selective fading channel. At the receiver end, after removing the guard interval and serial to parallel conversion, the sampled received signal $y_n$ becomes [22]

$$y_n = s_n \otimes h_n + v_n, \qquad n = 0,1.......N-1 \tag{4.3}$$

where $\otimes$ is convolution operation

$h_n$ is channel impulse response

$v_n$ is i.i.d complex white Gaussian noise

Let $\mathbf{h} = [h_0, h_1, ......, h_{L-1}]^T$ contain the time response of all $L$ taps of the channel, then frequency response of the channel $H_k$ is given by

$$H_k = \text{DFT}\{h_n\}$$

$$H_k = \sum_{n=0}^{L-1} h_n e^{-j2\pi kn/N}, \qquad k = 0,....,N-1 \tag{4.4}$$

By writing the equation (4.4) in matrix form, we get

$$H_k = \begin{bmatrix} 1 & e^{-j2\pi k/N} & \dots & e^{-j2\pi k(L-1)/N} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{L-1} \end{bmatrix} \tag{4.5}$$

Denote $\mathbf{w}_f(k) = \begin{bmatrix} 1 & e^{j2\pi k/N} & \dots & e^{j2\pi k(L-1)/N} \end{bmatrix}^T$, then equation (4.5) becomes

$$H_k = \mathbf{w}_f^H(k)\mathbf{h} \tag{4.6}$$

Received samples $\{y_n\}_{n=0}^{N-1}$ are applied to the Discrete Fourier Transform (DFT) block resulting in $\{Y_k\}_{k=0}^{N-1}$, which are given by

$$Y_k = \text{DFT}\{y_n\}$$

$$Y_k = \sum_{n=0}^{N-1} y_n e^{-j2\pi kn/N}; \qquad k = 0, \dots, N-1 \tag{4.7}$$

In frequency domain, by using equation (4.3) the received signal $Y_k$ over $k^{th}$ sub-carrier can be expressed as [34]

$$Y_k = S_k H_k + V_k, \qquad k = 0, 1, \dots\dots, N\text{-}1 \tag{4.8}$$

Where $V_k \sim N_c(0, \sigma^2)$ is the Fourier transform of $v_n$, which is also i.i.d. complex Gaussian noise Then received signal in equation (4.8) using equation(4.6) is given by

$$Y_k = S_k \mathbf{w}_f^H(k)\mathbf{h} + V_k, \qquad k = 0, 1, \dots\dots, N\text{-}1 \tag{4.9}$$

Assume that the channel response is constant during one OFDM symbol duration, then

$$\mathbf{h}_{k+1} = \mathbf{h}_k \tag{4.10}$$

Now the state space model of OFDM system in one symbol duration using equations (4.9) & (4.10) is given by

$$\mathbf{h}_{k+1} = \mathbf{h}_k \tag{4.11}$$

$$Y_k = S_k \mathbf{w}_f^H(k)\mathbf{h} + V_k \tag{4.12}$$

## 4.2 Particle Filtering Algorithm for OFDM System

Consider the state space model of OFDM system given by equations (4.11) & (4.12). Let $\mathbf{Y}_k \triangleq \{Y_0, Y_1, ........, Y_k\}$ be the received signal and $\mathbf{S}_k \triangleq \{S_0, S_1, ........, S_k\}$ be the transmitted signal up to $k^{th}$ sub-carrier respectively.

***Statement of the Problem*** : *To estimate the a posteriori probabilities of the information symbols*

$$p(d_k = a_i/\mathbf{Y}_k), \qquad a_i \in \mathbf{A}; k = 1, ...., N-1$$

*based on the received signal up to* $k^{th}$ sub-carrier $\mathbf{Y}_k$ *and the a priori symbol probabilities of* $d_k$ *without the knowledge of channel response* $\mathbf{h}_k$.

Consider that *M*-ary phase-shift keying (MPSK) signals are transmitted i.e.,

$$a_i = exp\left(j\frac{2\pi i}{|A|}\right), \qquad for \ \ i = 0, ........, |A|-1 \tag{4.13}$$

where $j = \sqrt{-1}$. Assume that the transmitted symbols are independent i.e.,

$$p(S_k = a_i/\mathbf{S}_{k-1}) = p(S_k = a_i), \qquad a_i \in A \tag{4.14}$$

Now, using equation (4.1) the probability $p(S_k = a_i)$ is given by

$$p(S_k = a_i) = p(d_k = a_i S_{k-1}^*) \tag{4.15}$$

When no prior information about the symbols is available, the symbols are assumed to take each possible value in the Alphabet set $A$ with equal probability i.e.,

$$p\left(d_k = a_i S^{*}_{k-1}\right) = \frac{1}{|A|} \quad \text{for } i = 1,.........|A| \tag{4.16}$$

In order to implement the particle filter, a set of Monte Carlo samples of the transmitted symbols $\left\{\mathbf{S}_k^{(j)}\right\}_{j=1}^{m}$ with its corresponding importance weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$ which are properly weighted with respect to the distribution $p\left(\mathbf{S}_k/\mathbf{Y}_k\right)$ are needed. From SIS method discussed in section 2.4.1, the Monte Carlo samples are easily generated from a trial sampling density. If the choice of trial sampling density is taken as the optimal sampling density then by equation (2.39)

$$q\left(S_k/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_k\right) = p\left(S_k/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_k\right) \tag{4.17}$$

For this choice of sampling density, the weights are updated according to equation (2.40) as

$$w_k^{(j)} \propto w_{k-1}^{(j)} p\left(Y_k/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) \tag{4.18}$$

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{a_i \in A} p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(S_k = a_i/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$$

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{a_i \in A} p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(d_k = a_i S_{k-1}^{(j)*}\right) \tag{4.19}$$

Denote $\alpha_{k,i}^{(j)} = p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(d_k = a_i S_{k-1}^{(j)*}\right)$, then weights are given by

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{a_i \in A} \alpha_{k,i}^{(j)} \tag{4.20}$$

The sampling density $p\left(S_k/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_k\right)$ can be written as

$$p\left(S_k/\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_k\right) = p\left(S_k/\mathbf{S}_{k-1}^{(j)}, Y_k, \mathbf{Y}_{k-1}\right)$$

$$= \frac{p\left(S_k, \mathbf{S}_{k-1}^{(j)}, Y_k, \mathbf{Y}_{k-1}\right)}{p\left(\mathbf{S}_{k-1}^{(j)}, Y_k, \mathbf{Y}_{k-1}\right)}$$

$$= \frac{p\left(Y_k / \mathbf{S}_{k-1}^{(j)}, S_k, \mathbf{Y}_{k-1}\right) p\left(\mathbf{S}_{k-1}^{(j)}, S_k, \mathbf{Y}_{k-1}\right)}{p\left(Y_k / \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(\mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)}$$

$$= \frac{p\left(Y_k / S_k, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(S_k / \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)}{p\left(Y_k / \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)}$$

$$\propto p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(S_k = a_i / \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$$

$$\propto p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(S_k = a_i\right)$$

$$\propto p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(d_k = a_i S_{k-1}^{(j)*}\right)$$

$$p\left(S_k / \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_k\right) = \alpha_{k,i}^{(j)} \qquad (4.21)$$

From the state space model (4.11) & (4.12), the density $p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$ is Gaussian and its mean and variance are calculated using the Kalman filtering algorithm.

$$p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) \sim N_c\left(mean, variance\right) \qquad (4.22)$$

The state space model of OFDM system defined by equations (4.11) & (4.12) is reproduced here.

$$\mathbf{h}_{k+1} = \mathbf{h}_k$$

$$Y_k = S_k \mathbf{w}_f^H(k)\mathbf{h} + V_k$$

The Kalman filtering algorithm for the above state space model when $S_k$ is known, is given below:

The innovation term from equation (2.17) is given by

$$\eta_k = Y_k - S_k \mathbf{w}_f^H(k)\hat{\mathbf{h}}_{k-1} \tag{4.23}$$

The correlation matrix of the innovation process by using equation (2.19) is given by

$$R_k = S_k \mathbf{w}_f^H(k)\mathbf{K}_k \mathbf{w}_f(k) S_k^* + \sigma^2$$

$$R_k = |S_k|^2 \mathbf{w}_f^H(k)\mathbf{K}_k \mathbf{w}_f(k) + \sigma^2$$

$$R_k = \mathbf{w}_f^H(k)\mathbf{K}_k \mathbf{w}_f(k) + \sigma^2 \qquad (\because |S_k| = 1) \tag{4.24}$$

The Kalman gain by using equation (2.20) is given by

$$\mathbf{g}_k = \mathbf{K}_k \mathbf{w}_f(k) S_k^* / R_k \tag{4.25}$$

Filtered state error correlation matrix by using equation (2.18) is given by

$$\mathbf{\Sigma}_k = \mathbf{K}_k - \mathbf{g}_k S_k \mathbf{w}_f^H(k)\mathbf{K}_k$$

$$\mathbf{\Sigma}_k = \mathbf{K}_k - \frac{1}{R_k}\mathbf{K}_k \mathbf{w}_f(k) S_k^* S_k \mathbf{w}_f^H(k)\mathbf{K}_k$$

$$\mathbf{\Sigma}_k = \mathbf{K}_k - \frac{1}{R_k}|S_k|^2 \mathbf{K}_k \mathbf{w}_f(k)\mathbf{w}_f^H(k)\mathbf{K}_k$$

$$\mathbf{\Sigma}_k = \mathbf{K}_k - \frac{1}{R_k}\mathbf{K}_k \mathbf{w}_f(k)\mathbf{w}_f^H(k)\mathbf{K}_k \qquad (\because |S_k| = 1) \tag{4.26}$$

Predicted state error correlation matrix by using equation (2.16) is given by

$$\mathbf{K}_k = \mathbf{\Sigma}_{k-1} \tag{4.27}$$

Substituting equation (4.27) in equations (4.24), (4.25) and (4.26) , we can write

$$R_k = \mathbf{w}_f^H(k)\boldsymbol{\Sigma}_{k-1}\mathbf{w}_f(k)+\sigma^2 \tag{4.28}$$

$$\mathbf{g}_k = \boldsymbol{\Sigma}_{k-1}\mathbf{w}_f(k)S_k^* / R_k \tag{4.29}$$

$$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k-1} - \frac{1}{R_k}\boldsymbol{\Sigma}_{k-1}\mathbf{w}_f(k)\mathbf{w}_f^H(k)\boldsymbol{\Sigma}_{k-1} \tag{4.30}$$

Estimated state vector by using equation (2.17) is given by

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \mathbf{g}_k\eta_k$$

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \frac{1}{R_k}\boldsymbol{\Sigma}_{k-1}\mathbf{w}_f(k)S_k^*\left(Y_k - S_k\mathbf{w}_f^H(k)\hat{\mathbf{h}}_{k-1}\right)$$

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \frac{\left(Y_k - S_k\mathbf{w}_f^H(k)\hat{\mathbf{h}}_{k-1}\right)}{R_k}\boldsymbol{\Sigma}_{k-1}\mathbf{w}_f(k)S_k^* \tag{4.31}$$

The *mean* of density $p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$ can be computed using the innovation equation (4.23) as [22]

$$mean = E\left\{Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right\} \tag{4.32}$$

$$= S_k\mathbf{w}_f^H(k)\mathbf{h}_{k-1}^{(j)}\Big|_{S_k=a_i}$$

$$= a_i\mathbf{w}_f^H(k)\mathbf{h}_{k-1}^{(j)}$$

Denoting $\mu_{k,i}^{(j)} = a_i\mathbf{w}_f^H(k)\mathbf{h}_{k-1}^{(j)}$, then

$$mean = \mu_{k,i}^{(j)} \tag{4.33}$$

The *variance* of density $p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$ can be computed by using the correlation matrix of the innovation process equation (4.28) as [22]

$$variance = \text{Var}\left\{Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right\} \tag{4.34}$$

$$= \mathbf{w}_f^H(k)\boldsymbol{\Sigma}_{k-1}^{(j)}\mathbf{w}_f(k) + \sigma^2\Big|_{S_k = a_i}$$

$$= \mathbf{w}_f^H(k)\boldsymbol{\Sigma}_{k-1}^{(j)}\mathbf{w}_f(k) + \sigma^2$$

which may be written as

$$variance = R_{k,i}^{(j)} \tag{4.35}$$

Now, the probability density $p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right)$ using equations (4.33) & (4.35) is given by

$$p\left(Y_k/S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) \sim N_c\left(\mu_{k,i}^{(j)}, R_{k,i}^{(j)}\right) \tag{4.36}$$

For each $a_i \in A$, the *a posteriori* symbol probability $p\left(d_k = a_i/\mathbf{Y}_k\right)$ as in equation (2.36) can be estimated as [22]

$$p\left(d_k = a_i/\mathbf{Y}_k\right) = p\left(S_k S_{k-1}^* = a_i/\mathbf{Y}_k\right)$$

$$= E\left\{\delta\left(S_k S_{k-1}^* = a_i\right)/\mathbf{Y}_k\right\}$$

$$\cong \frac{1}{W_k}\sum_{j=1}^{m}\delta\left(S_k^{(j)} S_{k-1}^{(j)*} = a_i\right)w_k^{(j)} \tag{4.37}$$

Where $W_k = \sum_{j=1}^{m} w_k^{(j)}$

$\delta(.)$ is dirac-delta function defined as

$$\delta\left(S_k^{(j)}S_{k-1}^{(j)*}=a_i\right)=\begin{cases}1, & \text{if} \quad S_k^{(j)}S_{k-1}^{(j)*}=a_i \\ 0, & \text{if} \quad S_k^{(j)}S_{k-1}^{(j)*}\neq a_i\end{cases} \tag{4.38}$$

The information symbol $d_k$ is estimated by maximizing the a posterior density $p(d_k = a_i/\mathbf{Y}_k)$, where $a_i \in A$. The particle filter algorithm for generating the sequential Monte Carlo samples of the transmitted symbols $\left\{\mathbf{S}_k^{(j)}\right\}_{j=1}^{m}$ with corresponding weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$, which are properly weighted with respect to the distribution $p(\mathbf{S}_k/\mathbf{Y}_k)$ and Kalman filter update $k_k^{(j)}=\left(\mathbf{h}_k^{(j)},\mathbf{\Sigma}_k^{(j)}\right)$ is given in algorithm 4.1[22].

***Algorithm 4.1***

1) *Initialization*

Each Kalman filter is initialized as $k_{-1}^{(j)}=\left(\mathbf{h}_{-1}^{(j)},\mathbf{\Sigma}_{-1}^{(j)}\right)$, with $\mathbf{\Sigma}_{-1}^{(j)}=1000\mathbf{I}_L$ and $\mathbf{h}_{-1}^{(j)} \sim N_c\left(\mathbf{0},\mathbf{\Sigma}_{-1}\right)$ for $j=1,.......m$. All the importance weights are initialized as $w_{-1}^{j}=1, j=1,.......m$ so that there is no bias in decision making by initial weights.

Based on the state space model (4.11) & (4.12), the following steps are implemented at $k^{th}$ recursion $(k=0,.......,N-1)$ to update each weighted sample. For $j=1,.......m$

2) *For each $a_i \in A$, compute the following quantities*

From equations (4.33) and (4.35), the mean and variance of the trial sampling density are calculated.

$$\mu_{k,i}^{(j)}=a_i\mathbf{w}_f^H(k)\mathbf{h}_{k-1}^{(j)}$$

$$R_{k,i}^{(j)}=\mathbf{w}_f^H(k)\mathbf{\Sigma}_{k-1}^{(j)}\mathbf{w}_f(k)+\sigma^2$$

3) *Compute the trial sampling density*

For each $a_i \in A$, compute $\alpha_{k,i}^{(j)}$ by using equation (4.36) as

$$\alpha_{k,i}^{(j)} = p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) p\left(d_k = a_i S_{k-1}^{(j)*}\right)$$

$$p\left(Y_k / S_k = a_i, \mathbf{S}_{k-1}^{(j)}, \mathbf{Y}_{k-1}\right) \sim N_c\left(\mu_{k,i}^{(j)}, R_{k,i}^{(j)}\right)$$

4) *Impute the symbol* $S_k$

Draw $S_k^{(j)}$ from the Alphabet set $A$ with probability

$$p(S_k^{(j)} = a_i) \propto \alpha_{k,i}^{(j)}, \quad a_i \in A \tag{4.39}$$

Append $S_k^{(j)}$ to $\mathbf{S}_{k-1}^{(j)}$ and obtain $\mathbf{S}_k^{(j)}$

5) *Compute the importance weight*

By using equation (4.20), the weight update is given by

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{a_i \in A} \alpha_{k,i}^{(j)}$$

6) *Update the a posteriori mean and covariance of channel:*

If the imputed symbol $S_k^{(j)} = a_i$ in step 4, then set $\mu_k^{(j)} = \mu_{k,i}^{(j)}$, $R_k^{(j)} = R_{k,i}^{(j)}$ and update the a posteriori mean and covariance of channel by using equations (4.31) & (4.30) as

$$\mathbf{h}_k^{(j)} = \mathbf{h}_{k-1}^{(j)} + \frac{\left(Y_k - \mu_k^{(j)}\right)}{R_k^{(j)}} \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{w}_f(k) S_k^{(j)*} \tag{4.40}$$

$$\mathbf{\Sigma}_k^{(j)} = \mathbf{\Sigma}_{k-1}^{(j)} - \frac{1}{R_k^{(j)}} \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{w}_f(k) \mathbf{w}_f^H(k) \mathbf{\Sigma}_{k-1}^{(j)} \tag{4.41}$$

At each recursion $k$, the dominant computation in this particle filtering algorithm involves the $m$ one-step Kalman filter updates for $\left(\mathbf{h}_k^{(j)}, \mathbf{\Sigma}_k^{(j)}\right)$. Since the $m$ samplers operate independently and in parallel, the SMC detector is well suited for parallel implementations.

The residual resampling algorithm with fixed resampling interval (that is, resampling is done every $k_0$ recursions), which forms a new set of weighted samples $\left\{\left(\tilde{\mathbf{S}}_k^{(j)}, \tilde{k}_k^{(j)}, \tilde{w}_k^{(j)}\right)\right\}_{j=1}^{m}$ from original set $\left\{\left(\mathbf{S}_k^{(j)}, k_k^{(j)}, w_k^{(j)}\right)\right\}_{j=1}^{m}$ can be generated as in algorithm 3.2.

## Delayed-weight estimation

In section 4.2, the problem of estimating the symbol $d_k$ based on the received signals $\mathbf{Y}_k$ up to $k^{th}$ sub-carrier by particle filtering algorithm is considered, which is similar to online filtering in dynamical systems. By the state space model (4.11) & (4.12), the future signals $\{\mathbf{Y}_i\}_{i=k+1}^{N-1}$ contain useful information about the channel. Hence, an estimation that is also based on these future observations is usually more accurate. In delayed-weight estimation, the inference on $d_k$ is made at later time $(k+\delta), \delta > 0$ with respect to the distribution $p(d_k/\mathbf{Y}_{k+\delta})$. If the set $\left\{\left(\mathbf{S}_k^{(j)}, w_k^{(j)}\right)\right\}_{j=1}^{m}$ is properly weighted with respect to $p(\mathbf{S}_k/\mathbf{Y}_k)$, then by induction, the set $\left\{\left(\mathbf{S}_{k+\delta}^{(j)}, w_{k+\delta}^{(j)}\right)\right\}_{j=1}^{m}$ is properly weighted with respect to $p(\mathbf{S}_{k+\delta}/\mathbf{Y}_{k+\delta}), \delta > 0$. Hence, by focussing on $\mathbf{S}_t$ at step $(k+\delta)$, the delayed estimate of the symbol can be obtained as [22]

$$p(d_k = a_i/\mathbf{Y}_{k+\delta}) \cong \frac{1}{W_{k+\delta}} \sum_{j=1}^{m} \delta\left(S_k^{(j)} S_{k-1}^{(j)*} = a_i\right) w_{k+\delta}^{(j)}, \qquad a_i \in A \qquad (4.42)$$

where $W_{k+\delta} = \sum_{j=1}^{m} w_{k+\delta}^{(j)}$. Since the weights $\left\{w_{k+\delta}^{(j)}\right\}_{j=1}^{m}$ contain the information about the signals $(Y_{k+1}, \ldots\ldots\ldots, Y_{k+\delta})$, the estimate in equation (4.42) is usually more accurate. The delayed estimation method incurs no additional computational cost (i.e., CPU time), but it requires some extra memory for storing $\left\{\left(S_{k+1}^{(j)},\ldots\ldots\ldots,S_{k+\delta}^{(j)}\right)\right\}_{j=1}^{m}$.

## 4.3 Simulation Results

For the simulation of blind detection in OFDM systems, the coefficients of time delay line (TDL) model of frequency-selective fading channels are assumed to be uncorrelated. All $L$ taps of fading channel are Rayleigh distributed and normalized such that total energy is equal to unity.

For the simulation of blind detection in OFDM system using particle filtering algorithm in MATLAB environment, the following parameters are used.

- Number of particles m= 50
- Modulation schemes: BPSK, QPSK
- Number of subcarriers $N$=64
- Number of taps of fading channel $L$=3
- Number of transmitted symbols=100000
- Delay in delayed weight method=5.
- Number of Monte Carlo simulations $N_m$=100
- Resampling is done at every $k_0 = 5$ recursions.

Steps carried out for the simulation of particle filtering algorithm for OFDM system are

1) Generate BPSK signals from alphabet set $A=\{-1,1\}$ randomly and differentially encode them before transmission using equation (4.1).
2) Generate the fading coefficients and observations according to state space model of OFDM system given by equations (4.11) & (4.12).
3) Generate the sequential Monte Carlo samples of transmitted symbols $\left\{S_k^{(j)}\right\}_{j=1}^{m}$ with corresponding importance weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$ at $k^{th}$ recursion by using the algorithm 4.1.

4) Do resampling by using algorithm (3.2) whenever $k$ is a multiple of $k_0$ else go back to step 3.
5) For each $a_i \in A$, the a posteriori symbol probability $p(d_k = a_i / \mathbf{Y}_k)$ is calculated using the equation (4.37).
6) The symbol is decoded by maximization of a posterior probability $p(d_k = a_i / \mathbf{Y}_k)$ and bit error rate (BER) is calculated between transmitted symbols and decoded symbols.

Steps from 1 to 6 are repeated for each independent Monte Carlo run and BER is averaged over all Monte Carlo runs. To measure the system performance, Word error rate (WER) is also calculated, which denotes the error rate of the whole data block transferred during one symbol duration.

Fig. 4.2 shows BER and WER performance of the particle filtering in blind detection of OFDM system with additive Gaussian noise and BPSK modulation. Besides this, the performance of MLSE receiver which serves as a lower bound is also plotted. It may be seen that the delayed weight method performance is close to the known channel bound. As SNR is varied from 0dB-30dB, the BER of delayed weight method decreases from 0.1345 to 0.00026 while BER of known channel bound decreases from 0.1295 to 0.0002. From SNR of 10dB-30dB, WER of delayed weight method shows significant improvement and in close to WER of the known channel bound. The WER of delayed weight method varies from 1.0001 to 0.0123 as SNR is varied from 10dB-30dB.

Fig. 4.3 shows BER and WER performance of the particle filtering in blind detection of OFDM system with additive Gaussian noise and QPSK modulation. For delayed weight method, the BER decreases from 0.3645 to 0.0016 and WER decreases from 1.0003 to 0.0624 as SNR is varied from 0dB-30dB. It may be seen that, the delayed weight method performance is close to the known channel bound. For instance at SNR=20dB, the delayed weight method gives BER of 0.0089 and known channel bound gives BER of 0.0055. The delayed weight method gives better performance for BPSK when compared to QPSK. For instance, the delayed weight method gives BER of 0.0053 for BPSK while BER of 0.0356 for QPSK at SNR=15dB. It is noted that WER of delayed weight method is also close to the known channel bound.

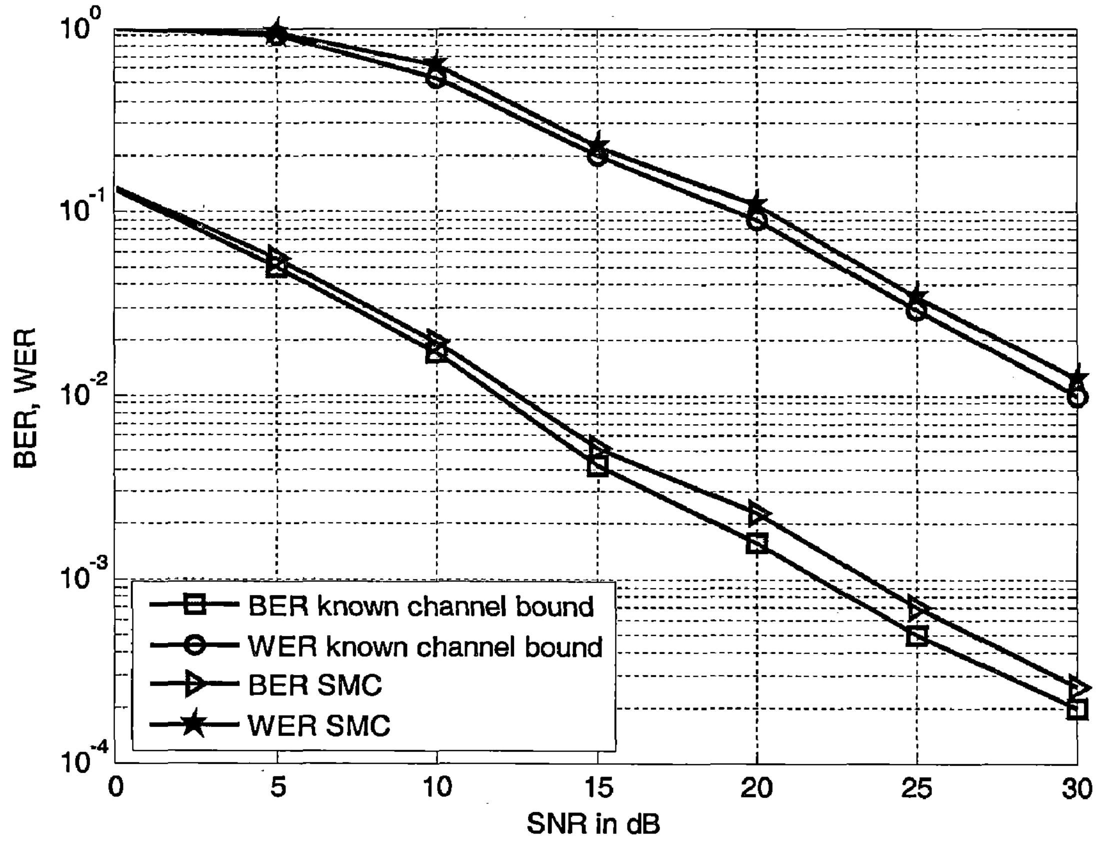



**Figure 4.2** BER and WER of OFDM system with BPSK modulation

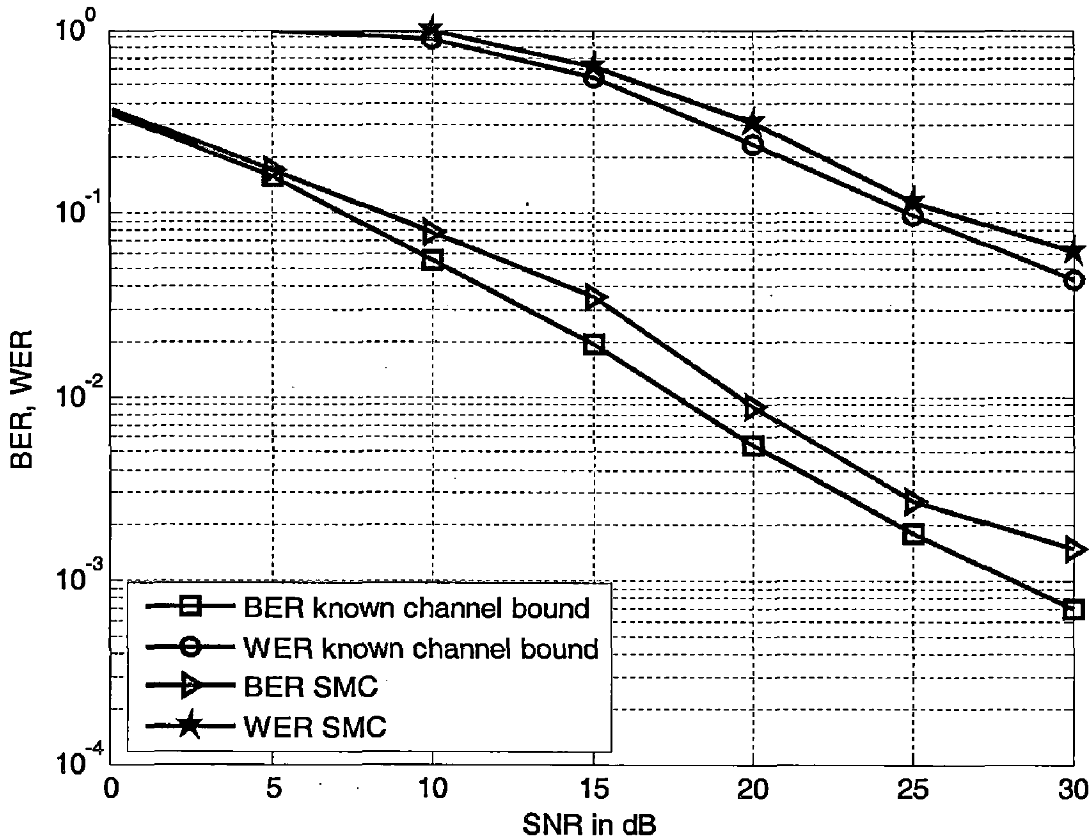



**Figure 4.3** BER and WER of OFDM system with QPSK modulation

# Chapter 5

# PARTICLE FILTERING FOR BLIND DETECTION IN MIMO-OFDM SYSTEMS

MIMO system is used for high data rate transmission in a dense multi-path scattering environment, which causes the MIMO channel to be frequency-selective. OFDM can transform such a frequency-selective MIMO channel into a set of parallel frequency-flat MIMO channels, and it reduces the ISI caused by the multipath. Thus MIMO-OFDM systems are used as effective means of providing high-speed data transmission over dispersive wireless channels [23]. For MIMO-OFDM systems, efficient and accurate channel estimation is necessary to coherently demodulate the received data. Jiang Yue *et al.*, [36] propose joint semi-blind channel estimation and data detection in MIMO-OFDM systems, where channel estimation is done by the Kalman filter by the use of pilots and the outputs of data detection are fed back to another Kalman filter for improved channel estimation. A sequential Monte Carlo Kalman filter based Channel estimation in MIMO-OFDM systems is presented in [37], where the initial estimate for the delays and channels are obtained using one training symbol and in the remaining OFDM symbol intervals, the QRD-M algorithm is employed to detect the transmitted data symbols and finally the QRD-M data detector and MC channel/delay estimator are combined in a joint decision-directed algorithm. The QRD-M algorithm [37,38] uses a limited tree search to approximate the maximum-likelihood detector.

We have used the state space model approach for deriving the particle filtering algorithm for the blind detection in differentially encoded MIMO-OFDM systems with the use of Kalman filtering algorithm. 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, the MIMO-OFDM system is described first. Next, the state space model of MIMO-OFDM system is derived. The particle filtering algorithm for the blind detection in differentially encoded MIMO-OFDM system is derived. Finally the simulation results are presented.

## 5.1 Signal Model of MIMO-OFDM System

Consider a MIMO-OFDM system with $P$ transmit antennas, $Q$ receive antennas and $N$ data sub-carriers as shown in Fig. 5.1.

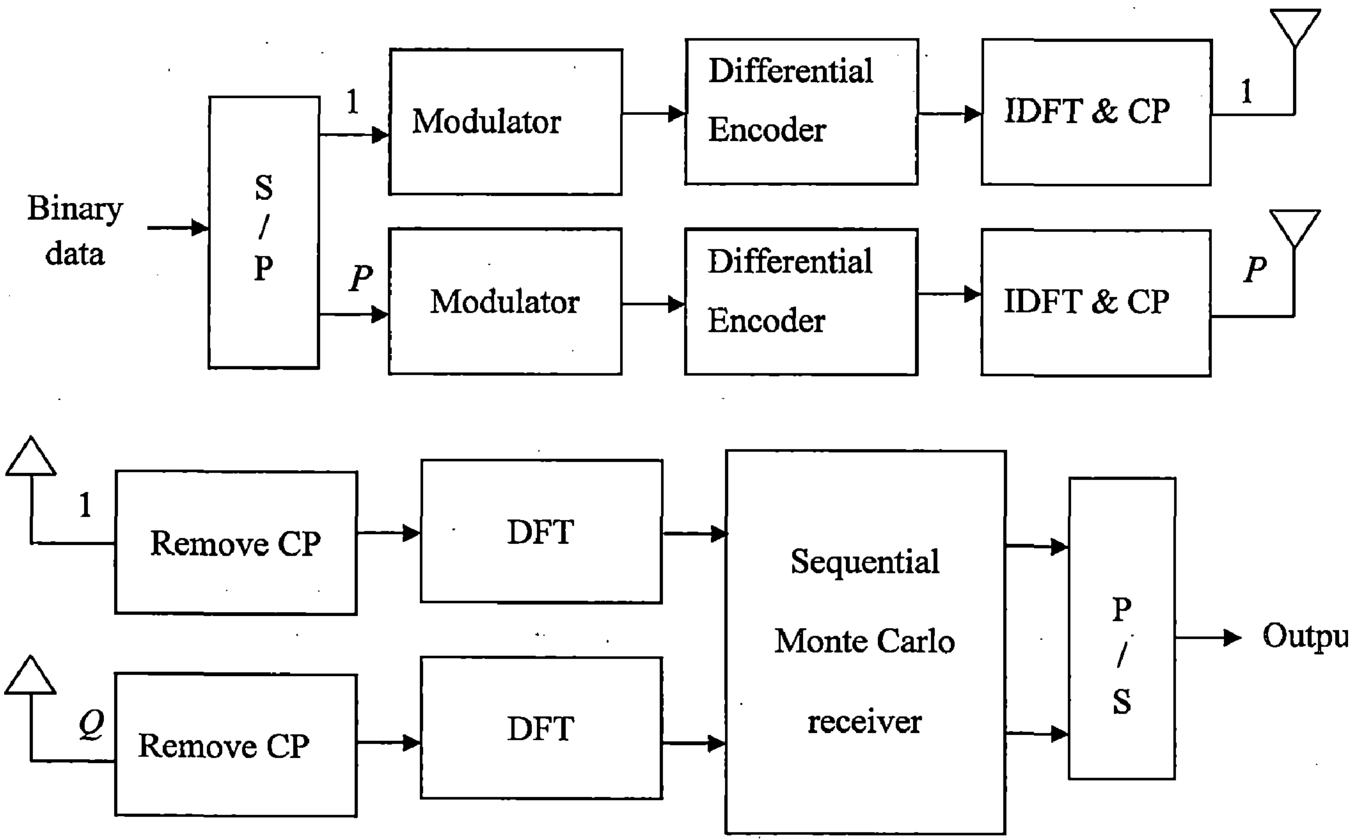



**Figure 5.1** An MIMO-OFDM system with sequential Monte Carlo receiver

As shown in Fig. 5.1, the binary data is demultiplexed into $P$ parallel streams; data in each stream is independently mapped into equiprobable symbols $d_k^p, p=1,\ldots\ldots P$ and $k=0,\ldots\ldots N-1$ within the modulator. These symbols $d_k^p$ are differentially encoded to resolve the phase ambiguity inherent to any blind receiver, and the output of differential encoder $S_k^p$ for each transmit antenna $p$ and sub-carrier $k$ is given by

$$\begin{aligned} S_0^p &= 1, \\ S_k^p &= S_{k-1}^p d_k^p, \qquad k=1,\ldots,N-1, \qquad p=1,\ldots\ldots P \end{aligned} \tag{5.1}$$

These differentially encoded symbols $\{S_k^p\}_{k=0}^{N-1}$, $p=1,\ldots\ldots,P$ are applied to the Inverse Discrete Fourier Transform (IDFT) block resulting in $\{s_n^p\}_{n=0}^{N-1}, p=1,\ldots\ldots,P$, which are given by

$$s_n^p = \text{IDFT}\{S_k^p\}$$

$$s_n^p = \frac{1}{N}\sum_{k=0}^{N-1} S_k^p e^{j2\pi kn/N}, \qquad n=0,\ldots.,N-1; \qquad p=1,\ldots\ldots P \tag{5.2}$$

The symbols $\{s_n^p\}_{n=0}^{N-1}, p=1,\ldots\ldots,P$ are transmitted via respective antenna $p$ after insertion of cyclic prefix (CP) in each OFDM symbol. At the receiver end, after removing CP and taking DFT then OFDM signal $Y_k^q$ received at the $q^{th}$ receive antenna at sub-carrier $k$ is given by [36,37]

$$Y_k^q = \sum_{p=1}^{P} H_k^{q,p} S_k^p + V_k^q, \qquad k=0,\ldots\ldots,N-1,\ p=1,\ldots\ldots,P,\ q=1,\ldots\ldots,Q \tag{5.3}$$

where $S_k^p$ is the transmitted symbol at the $p^{th}$ transmit antenna at sub-carrier $k$

$H_k^{q,p}$ is the frequency response of the channel between the $p^{th}$ transmit antenna and the $q^{th}$ receive antenna at sub-carrier $k$

$V_k^q$ is i.i.d complex Gaussian noise $\left(V_k^q \sim N_c\left(0,\sigma^2\right)\right)$

Let $\mathbf{h}^{q,p} = [h_0^{q,p}, h_1^{q,p}, \ldots\ldots., h_{L-1}^{q,p}]^T$ contains the time response of all $L$ taps of the channel between the $p^{th}$ transmit antenna and the $q^{th}$ receive antenna then frequency response of the channel $H_k^{q,p}$ is given by

$$H_k^{q,p} = \text{DFT}\{h_n^{q,p}\}$$

$$H_k^{q,p} = \sum_{n=0}^{L-1} h_n^{q,p} e^{-j2\pi kn/N}, \qquad k=0,\ldots\ldots,N\text{-}1,\ p=1,\ldots\ldots,P,\ q=1,\ldots\ldots,Q \tag{5.4}$$

By writing the equation (5.4) in matrix form, we get

$$H_k^{q,p} = \begin{bmatrix} 1 & e^{-j2\pi k/N} & \cdots & e^{-j2\pi k(L-1)/N} \end{bmatrix} \begin{bmatrix} h_0^{q,p} \\ h_1^{q,p} \\ \vdots \\ h_{L-1}^{q,p} \end{bmatrix} \tag{5.5}$$

Denote $\mathbf{w}_f(k) = \begin{bmatrix} 1 & e^{j2\pi k/N} & \cdots & e^{j2\pi k(L-1)/N} \end{bmatrix}^T$, then equation (5.5) becomes

$$H_k^{q,p} = \mathbf{w}_f^H(k)\mathbf{h}^{q,p}, \qquad k = 0,......,N\text{-}1,\ p = 1,......,P,\ q = 1,......,Q \tag{5.6}$$

The equation (5.6) for $q$=1 is written in matrix-form as

$$\begin{bmatrix} H_k^{1,1} \\ H_k^{1,2} \\ \vdots \\ H_k^{1,P} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_f^H(k) & 0 & \cdots & 0 \\ 0 & \mathbf{w}_f^H(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{w}_f^H(k) \end{bmatrix} \begin{bmatrix} \mathbf{h}^{1,1} \\ \mathbf{h}^{1,2} \\ \vdots \\ \mathbf{h}^{1,P} \end{bmatrix} \tag{5.7}$$

Let,

$$\begin{aligned} \mathbf{H}_k^1 &= \begin{bmatrix} H_k^{1,1} & H_k^{1,2} & \cdots & H_k^{1,P} \end{bmatrix}^T \\ \mathbf{H}_k^2 &= \begin{bmatrix} H_k^{2,1} & H_k^{2,2} & \cdots & H_k^{2,P} \end{bmatrix}^T \\ &\vdots \\ \mathbf{H}_k^Q &= \begin{bmatrix} H_k^{Q,1} & H_k^{Q,1} & \cdots & H_k^{Q,P} \end{bmatrix}^T \end{aligned}$$

$$\mathbf{C}_k = \begin{bmatrix} \mathbf{w}_f^H(k) & 0 & \cdots & 0 \\ 0 & \mathbf{w}_f^H(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{w}_f^H(k) \end{bmatrix}$$

and $\mathbf{S}_k = \begin{bmatrix} S_k^1 & S_k^2 & \cdots & S_k^P \end{bmatrix}$ then equation (5.3) in matrix- form is written as

$$\begin{bmatrix} Y_k^1 \\ Y_k^2 \\ \vdots \\ Y_k^Q \end{bmatrix} = \begin{bmatrix} \mathbf{S}_k & 0 & \cdots & 0 \\ 0 & \mathbf{S}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}_k \end{bmatrix} \begin{bmatrix} \mathbf{H}_k^1 \\ \mathbf{H}_k^2 \\ \vdots \\ \mathbf{H}_k^Q \end{bmatrix} + \begin{bmatrix} V_k^1 \\ V_k^2 \\ \vdots \\ V_k^Q \end{bmatrix} \tag{5.8}$$

Let,

$$\mathbf{h}^1 = \begin{bmatrix} \mathbf{h}^{1,1} & \mathbf{h}^{1,2} & \cdots & \mathbf{h}^{1,P} \end{bmatrix}^T$$
$$\mathbf{h}^2 = \begin{bmatrix} \mathbf{h}^{2,1} & \mathbf{h}^{2,2} & \cdots & \mathbf{h}^{2,P} \end{bmatrix}^T$$
$$\vdots$$
$$\mathbf{h}^Q = \begin{bmatrix} \mathbf{h}^{Q,1} & \mathbf{h}^{Q,2} & \cdots & \mathbf{h}^{Q,P} \end{bmatrix}^T$$

then equation (5.6) in matrix-form is written as

$$\begin{bmatrix} \mathbf{H}_k^1 \\ \mathbf{H}_k^2 \\ \vdots \\ \mathbf{H}_k^Q \end{bmatrix} = \begin{bmatrix} \mathbf{C}_k & 0 & \cdots & 0 \\ 0 & \mathbf{C}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \vdots \\ \mathbf{h}^Q \end{bmatrix} \tag{5.9}$$

By substituting equation (5.9) in equation (5.8), we get

$$\begin{bmatrix} Y_k^1 \\ Y_k^2 \\ \vdots \\ Y_k^Q \end{bmatrix} = \begin{bmatrix} \mathbf{S}_k & 0 & \cdots & 0 \\ 0 & \mathbf{S}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}_k \end{bmatrix} \begin{bmatrix} \mathbf{C}_k & 0 & \cdots & 0 \\ 0 & \mathbf{C}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \vdots \\ \mathbf{h}^Q \end{bmatrix} + \begin{bmatrix} V_k^1 \\ V_k^2 \\ \vdots \\ V_k^Q \end{bmatrix} \tag{5.10}$$

Denote,

$$\mathbf{\Psi}_k = \begin{bmatrix} \mathbf{S}_k & 0 & \cdots & 0 \\ 0 & \mathbf{S}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}_k \end{bmatrix}$$

$$\mathbf{\Omega}_k = \begin{bmatrix} \mathbf{C}_k & 0 & \cdots & 0 \\ 0 & \mathbf{C}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_k \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}^1 & \mathbf{h}^2 & \cdots & \mathbf{h}^Q \end{bmatrix}^T$$

$$\mathbf{Y}_k = \begin{bmatrix} Y_k^1 & Y_k^2 & \cdots & Y_k^Q \end{bmatrix}^T$$

$$\mathbf{V}_k = \begin{bmatrix} V_k^1 & V_k^2 & \cdots & V_k^Q \end{bmatrix}^T$$

The equation (5.10) is modified as

$$\mathbf{Y}_k = \mathbf{\Psi}_k \mathbf{\Omega}_k \mathbf{h} + \mathbf{V}_k \tag{5.11}$$

Assume that the channel is static in one OFDM symbol duration, then

$$\mathbf{h}_{k+1} = \mathbf{h}_k \tag{5.12}$$

The state space model of MIMO-OFDM system by using equations (5.11) & (5.12) is given by

$$\mathbf{h}_{k+1} = \mathbf{h}_k \tag{5.13}$$

$$\mathbf{Y}_k = \mathbf{\Psi}_k \mathbf{\Omega}_k \mathbf{h}_k + \mathbf{V}_k \tag{5.14}$$

## 5.2 Particle Filtering Algorithm for MIMO-OFDM System

Consider the state space model of MIMO-OFDM system given by equations (5.13) & (5.14). Let $\mathcal{Y}_k \triangleq \{\mathbf{Y}_0, \mathbf{Y}_1, \ldots\ldots\ldots, \mathbf{Y}_k\}$ be the received signal and $\mathcal{S}_k \triangleq \{\mathbf{S}_0, \mathbf{S}_1, \ldots\ldots\ldots, \mathbf{S}_k\}$ be the transmitted signal up to $k^{th}$ sub-carrier respectively. Also denote $\mathbf{d}_k \triangleq \{d_k^1, d_k^2, \ldots\ldots\ldots, d_k^P\}$ and $\mathcal{D}_k \triangleq \{\mathbf{d}_1, \mathbf{d}_2, \ldots\ldots\ldots, \mathbf{d}_k\}$.

***Statement of the Problem:*** *To estimate the a posteriori probabilities of the information symbols*

$$p\left(\mathbf{d}_k = \mathbf{a}_i / \mathcal{Y}_k\right), \qquad \mathbf{a}_i \in A^P; k = 1, \ldots, N-1 \tag{5.15}$$

*based on the received signal* $\mathcal{Y}_k$ *up to* $k^{th}$ *sub-carrier and the a priori symbol probabilities of* $\mathcal{D}_k$ *without the knowledge of channel response* $\mathbf{h}_k$.

Consider that $M$-ary phase-shift keying (MPSK) signals are transmitted i.e.,

$$a_i = exp\left(j\frac{2\pi i}{|A|}\right), \qquad for \;\; i = 0, \ldots\ldots, |A|-1 \tag{5.16}$$

where $j = \sqrt{-1}$. Assume that the transmitted symbols are independent i.e.,

$$p\left(\mathbf{S}_k = \mathbf{a}_i / \mathcal{S}_{k-1}\right) = p\left(\mathbf{S}_k = \mathbf{a}_i\right), \qquad \mathbf{a}_i \in A^P \tag{5.17}$$

Now, using equation (5.1) the probability $p\left(\mathbf{S}_k = \mathbf{a}_i\right)$ is given by

$$p\left(\mathbf{S}_k = \mathbf{a}_i\right) = p\left(\mathbf{d}_k = \mathbf{a}_i \circ \mathbf{S}_{k-1}^*\right) \tag{5.18}$$

where $\circ$ denotes element-wise product. When no prior information about the symbols is available, the symbols are assumed to take each possible value in the Alphabet set $A^P$ with equal probability i.e.,

$$p\left(\mathbf{d}_k = \mathbf{a}_i \circ \mathbf{S}_{k-1}^*\right) = \frac{1}{|A^P|} \qquad \text{for} \;\; i = 1, \ldots\ldots\ldots |A^P| \tag{5.19}$$

Let $\mathbf{S}_k^{(j)} \triangleq \left\{S_k^{1(j)}, S_k^{2(j)}, \ldots\ldots\ldots, S_k^{P(j)}\right\}, j = 1, 2, \ldots\ldots, m$ be a sample drawn at sub-carrier $k$ and denote $\mathcal{S}_k^{(j)} \triangleq \left\{\mathbf{S}_0^{(j)}, \mathbf{S}_1^{(j)}, \ldots\ldots\ldots, \mathbf{S}_k^{(j)}\right\}$. In order to implement the particle filter, a set of Monte Carlo samples of the transmitted symbols $\left\{\mathcal{S}_k^{(j)}\right\}_{j=1}^{m}$ with its corresponding importance weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$ which are properly weighted with respect to the distribution $p(\mathcal{S}_k/\mathcal{Y}_k)$ are needed. From SIS method discussed in section 2.4.1, the Monte Carlo samples are easily generated from a trial sampling density. If the choice of trial sampling density is taken as the optimal sampling density then by equation (2.39), we get

$$q\left(\mathbf{S}_k/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_k\right)=p\left(\mathbf{S}_k/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_k\right) \tag{5.20}$$

For this choice of sampling density, the weights are updated according to equation (2.40) as

$$w_k^{(j)} \propto w_{k-1}^{(j)} p\left(\mathbf{Y}_k/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_{k-1}\right) \tag{5.21}$$

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{\mathbf{a}_i\in A^P} p\left(\mathbf{Y}_k/\mathcal{S}_{k-1}^{(j)},\mathbf{S}_k=\mathbf{a}_i,\mathcal{Y}_{k-1}\right)p\left(\mathbf{S}_k=\mathbf{a}_i/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_{k-1}\right)$$

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{\mathbf{a}_i\in A^P} p\left(\mathbf{Y}_k/\mathcal{S}_{k-1}^{(j)},\mathbf{S}_k=\mathbf{a}_i,\mathcal{Y}_{k-1}\right)p\left(\mathbf{S}_k=\mathbf{a}_i\right)$$

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{\mathbf{a}_i\in A^P} p\left(\mathbf{Y}_k/\mathcal{S}_{k-1}^{(j)},\mathbf{S}_k=\mathbf{a}_i,\mathcal{Y}_{k-1}\right)p\left(\mathbf{d}_k=\mathbf{a}_i\circ\mathbf{S}_{k-1}^{(j)*}\right) \tag{5.22}$$

Denote $\alpha_{k,i}^{(j)}=p\left(\mathbf{Y}_k/\mathcal{S}_{k-1}^{(j)},\mathbf{S}_k=\mathbf{a}_i,\mathcal{Y}_{k-1}\right)p\left(\mathbf{d}_k=\mathbf{a}_i\circ\mathbf{S}_{k-1}^{(j)*}\right)$, then weights are given by

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{\mathbf{a}_i\in A^P} \alpha_{k,i}^{(j)} \tag{5.23}$$

The sampling density $p\left(\mathbf{S}_k/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_k\right)$ can be written as

$$p\left(\mathbf{S}_k/\mathcal{S}_{k-1}^{(j)},\mathcal{Y}_k\right)=p\left(\mathbf{S}_k/\mathcal{S}_{k-1}^{(j)},\mathbf{Y}_k,\mathcal{Y}_{k-1}\right)$$

$$=\frac{p\left(\mathbf{S}_k,\mathcal{S}_{k-1}^{(j)},\mathbf{Y}_k,\mathcal{Y}_{k-1}\right)}{p\left(\mathcal{S}_{k-1}^{(j)},\mathbf{Y}_k,\mathcal{Y}_{k-1}\right)}$$

$$= \frac{p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k, \mathcal{Y}_{k-1}\right) p\left(\mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k, \mathcal{Y}_{k-1}\right)}{p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_{k-1}\right) p\left(\mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_{k-1}\right)}$$

$$= \frac{p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k, \mathcal{Y}_{k-1}\right) p\left(\mathbf{S}_k / \mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_{k-1}\right)}{p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_{k-1}\right)}$$

$$\propto p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) p\left(\mathbf{S}_k = \mathbf{a}_i / \mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_{k-1}\right)$$

$$\propto p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) p\left(\mathbf{S}_k = \mathbf{a}_i\right)$$

$$\propto p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) p\left(\mathbf{d}_k = \mathbf{a}_i \circ \mathbf{S}_{k-1}^{(j)*}\right)$$

$$p\left(\mathbf{S}_k / \mathcal{S}_{k-1}^{(j)}, \mathcal{Y}_k\right) = \alpha_{k,i}^{(j)} \tag{5.24}$$

From the state space model (5.13) & (5.14), the density $p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right)$ is Gaussian and its mean and variance are calculated using the Kalman filtering algorithm.

$$p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) \sim N_c\left(mean, variance\right) \tag{5.25}$$

The state space model of MIMO–OFDM system defined by equations (5.13) & (5.14) is reproduced here.

$$\mathbf{h}_{k+1} = \mathbf{h}_k$$

$$\mathbf{Y}_k = \mathbf{\Psi}_k \mathbf{\Omega}_k \mathbf{h}_k + \mathbf{V}_k$$

The Kalman filtering algorithm for the above state space model when $\mathbf{S}_k$ is known, is given as follows:

The innovation term from equation (2.17) is given by

$$\boldsymbol{\eta}_k = \mathbf{Y}_k - \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \hat{\mathbf{h}}_{k-1} \tag{5.26}$$

The correlation matrix of the innovation process by using equation (2.19) is given by

$$\mathbf{R}_k = \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \mathbf{K}_k \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H + \sigma^2 \mathbf{I}_Q \tag{5.27}$$

The Kalman gain by using equation (2.20) is given by

$$\mathbf{G}_k = \mathbf{K}_k \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H \mathbf{R}_k^{-1} \tag{5.28}$$

Filtered state error correlation matrix by using equation (2.18) is given by

$$\boldsymbol{\Sigma}_k = \mathbf{K}_k - \mathbf{G}_k \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \mathbf{K}_k$$

$$\boldsymbol{\Sigma}_k = \mathbf{K}_k - \mathbf{K}_k \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H \mathbf{R}_k^{-1} \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \mathbf{K}_k \tag{5.29}$$

Predicted state error correlation matrix by using equation (2.16) is given by

$$\mathbf{K}_k = \boldsymbol{\Sigma}_{k-1} \tag{5.30}$$

Substituting equation (5.30) in equations (5.27), (5.28) and (5.29), we can write,

$$\mathbf{R}_k = \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H + \sigma^2 \mathbf{I}_Q \tag{5.31}$$

$$\mathbf{G}_k = \boldsymbol{\Sigma}_{k-1} \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H \mathbf{R}_k^{-1} \tag{5.32}$$

$$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k-1} - \boldsymbol{\Sigma}_{k-1} \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H \mathbf{R}_k^{-1} \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \boldsymbol{\Sigma}_{k-1} \tag{5.33}$$

Estimated state vector by using equation (2.17) is given by

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \mathbf{G}_k \boldsymbol{\eta}_k$$

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \boldsymbol{\Sigma}_{k-1} \boldsymbol{\Omega}_k^H \boldsymbol{\Psi}_k^H \mathbf{R}_k^{-1} \left( \mathbf{Y}_k - \boldsymbol{\Psi}_k \boldsymbol{\Omega}_k \hat{\mathbf{h}}_{k-1} \right) \tag{5.34}$$

The *mean* of density $p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right)$ can be computed using the innovation equation (5.26) as

$$mean = E\left\{\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right\} \tag{5.35}$$

$$= \mathbf{\Psi}_k \mathbf{\Omega}_k \mathbf{h}_{k-1}^{(j)}\Big|_{\mathbf{S}_k = \mathbf{a}_i}$$

$$= \mathbf{\Psi}\left(\mathbf{a}_i\right) \mathbf{\Omega}_k \mathbf{h}_{k-1}^{(j)}$$

Denoting $\mathbf{\Xi}_i = \mathbf{\Psi}\left(\mathbf{a}_i\right)$, then

$$mean = \mathbf{\Xi}_i \mathbf{\Omega}_k \mathbf{h}_{k-1}^{(j)} \tag{5.36}$$

Let $\boldsymbol{\mu}_{k,i}^{(j)} = \mathbf{\Xi}_i \mathbf{\Omega}_k \mathbf{h}_{k-1}^{(j)}$ then *mean* is given by

$$mean = \boldsymbol{\mu}_{k,i}^{(j)} \tag{5.37}$$

The *variance* of density $p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right)$ can be computed by using the correlation matrix of the innovation process given in equation (5.31) as

$$variance = \mathrm{Var}\left\{\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right\} \tag{5.38}$$

$$= \mathbf{\Psi}_k \mathbf{\Omega}_k \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{\Omega}_k^H \mathbf{\Psi}_k^H + \sigma^2 \mathbf{I}_Q\Big|_{\mathbf{S}_k = \mathbf{a}_i}$$

$$= \mathbf{\Psi}\left(\mathbf{a}_i\right) \mathbf{\Omega}_k \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{\Omega}_k^H \mathbf{\Psi}^H\left(\mathbf{a}_i\right) + \sigma^2 \mathbf{I}_Q$$

$$variance = \mathbf{\Xi}_i \mathbf{\Omega}_k \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{\Omega}_k^H \mathbf{\Xi}_i^H + \sigma^2 \mathbf{I}_Q \tag{5.39}$$

Let $\mathbf{R}_{k,i}^{(j)} = \mathbf{\Xi}_i \mathbf{\Omega}_k \mathbf{\Sigma}_{k-1}^{(j)} \mathbf{\Omega}_k^H \mathbf{\Xi}_i^H + \sigma^2 \mathbf{I}_Q$ then *variance* is given by

$$variance = \mathbf{R}_{k,i}^{(j)} \tag{5.40}$$

The probability density $p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right)$ using equations (5.37) & (5.40) is given by

$$p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) \sim N_c\left(\boldsymbol{\mu}_{k,i}^{(j)}, \mathbf{R}_{k,i}^{(j)}\right) \tag{5.41}$$

For each $\mathbf{a}_i \in A^P$, the *a posteriori* symbol probability $p\left(\mathbf{d}_k = \mathbf{a}_i / \mathcal{Y}_k\right)$ as in equation (2.36) can be estimated as

$$\begin{aligned} p\left(\mathbf{d}_k = \mathbf{a}_i / \mathcal{Y}_k\right) &= p\left(\mathbf{S}_k \circ \mathbf{S}_{k-1}^* = \mathbf{a}_i / \mathcal{Y}_k\right) \\ &= E\left\{\delta\left(\mathbf{S}_k \circ \mathbf{S}_{k-1}^* = \mathbf{a}_i\right) / \mathcal{Y}_k\right\} \\ &\cong \frac{1}{W_k} \sum_{j=1}^{m} \delta\left(\mathbf{S}_k^{(j)} \circ \mathbf{S}_{k-1}^{(j)*} = \mathbf{a}_i\right) w_k^{(j)} \end{aligned} \tag{5.42}$$

where $W_k = \sum_{j=1}^{m} w_k^{(j)}$

$\delta(.)$ is dirac-delta function defined as

$$\delta\left(\mathbf{S}_k^{(j)} \circ \mathbf{S}_{k-1}^{(j)*} = \mathbf{a}_i\right) = \begin{cases} 1, & \text{if} \quad \mathbf{S}_k^{(j)} \circ \mathbf{S}_{k-1}^{(j)*} = \mathbf{a}_i \\ 0, & \text{if} \quad \mathbf{S}_k^{(j)} \circ \mathbf{S}_{k-1}^{(j)*} \neq \mathbf{a}_i \end{cases} \tag{5.43}$$

The information symbol $\mathbf{d}_k$ is estimated by maximizing the a posterior density $p\left(\mathbf{d}_k = \mathbf{a}_i / \mathcal{Y}_k\right)$, where $\mathbf{a}_i \in A^P$. The particle filter algorithm for generating the sequential Monte Carlo samples of the transmitted symbols $\left\{\mathcal{S}_k^{(j)}\right\}_{j=1}^{m}$ with corresponding weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$, which are properly weighted with respect to the distribution $p\left(\mathcal{S}_k / \mathcal{Y}_k\right)$ and Kalman filter update $\kappa_k^{(j)} = \left(\mathbf{h}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)}\right)$ is given in algorithm 5.1.

*Algorithm 5.1*

1) *Initialization*

Each Kalman filter is initialized as $k_{-1}^{(j)} = \left(\mathbf{h}_{-1}^{(j)}, \boldsymbol{\Sigma}_{-1}^{(j)}\right)$, with $\boldsymbol{\Sigma}_{-1}^{(j)} = 1000\mathbf{I}_{LPQ}$ and $\mathbf{h}_{-1}^{(j)} \sim N_c\left(\mathbf{0}, \boldsymbol{\Sigma}_{-1}\right)$ for $j = 1, \ldots\ldots m$. All the importance weights are initialized as $w_{-1}^{j} = 1, j = 1, \ldots\ldots m$ so that there is no bias in decision making by initial weights.

Based on the state space model (5.13) & (5.14), the following steps are implemented at $k^{th}$ recursion $(k = 0, \ldots\ldots, N-1)$ to update each weighted sample. For $j = 1, \ldots\ldots m$

2) *For each* $\mathbf{a}_i \in A^P$, *compute the following quantities*

From equations (5.37) and (5.40), the mean and variance of the trial sampling density are calculated.

$$\boldsymbol{\mu}_{k,i}^{(j)} = \boldsymbol{\Xi}_i \boldsymbol{\Omega}_k \mathbf{h}_{k-1}^{(j)}$$

$$\mathbf{R}_{k,i}^{(j)} = \boldsymbol{\Xi}_i \boldsymbol{\Omega}_k \boldsymbol{\Sigma}_{k-1}^{(j)} \boldsymbol{\Omega}_k^H \boldsymbol{\Xi}_i^H + \sigma^2 \mathbf{I}_Q$$

where $\boldsymbol{\Xi}_i = \boldsymbol{\Psi}\left(\mathbf{a}_i\right)$.

3) *Compute the trial sampling density*

For each $\mathbf{a}_i \in A^P$, compute $\alpha_{k,i}^{(j)}$ by using equation (5.41) as

$$\alpha_{k,i}^{(j)} = p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) p\left(\mathbf{d}_k = \mathbf{a}_i \circ \mathbf{S}_{k-1}^{(j)*}\right).$$

$$p\left(\mathbf{Y}_k / \mathcal{S}_{k-1}^{(j)}, \mathbf{S}_k = \mathbf{a}_i, \mathcal{Y}_{k-1}\right) \sim N_c\left(\boldsymbol{\mu}_{k,i}^{(j)}, \mathbf{R}_{k,i}^{(j)}\right)$$

4) *Impute the symbol* $\mathbf{S}_k$

Draw a sample $\mathbf{S}_k^{(j)}$ from the set $A^P$ with probability

$$p(\mathbf{S}_k^{(j)} = \mathbf{a}_i) \propto \alpha_{k,i}^{(j)}, \qquad \mathbf{a}_i \in A^P \tag{5.44}$$

Append $\mathbf{S}_k^{(j)}$ to $\mathcal{S}_{k-1}^{(j)}$ and obtain $\mathcal{S}_k^{(j)}$

5) *Compute the importance weight*

By using equation (5.23), the weight update is given by

$$w_k^{(j)} \propto w_{k-1}^{(j)} \sum_{\mathbf{a}_i \in A^P} \alpha_{k,i}^{(j)}$$

6) *Update the a posteriori mean and covariance of channel*

If the imputed symbol $\mathbf{S}_k^{(j)} = \mathbf{a}_i$ in step 4, then set $\boldsymbol{\mu}_k^{(j)} = \boldsymbol{\mu}_{k,i}^{(j)}$, $\mathbf{R}_k^{(j)} = \mathbf{R}_{k,i}^{(j)}$ and update the a posteriori mean and covariance of channel by using equations (5.34) & (5.33) as

$$\mathbf{h}_k^{(j)} = \mathbf{h}_{k-1}^{(j)} + \boldsymbol{\Sigma}_{k-1}^{(j)} \boldsymbol{\Omega}_k^H \boldsymbol{\Xi}_k^{(j)H} \left(\mathbf{R}_k^{(j)}\right)^{-1} \left(\mathbf{Y}_k - \boldsymbol{\mu}_k^{(j)}\right) \tag{5.45}$$

$$\boldsymbol{\Sigma}_k^{(j)} = \boldsymbol{\Sigma}_{k-1}^{(j)} - \boldsymbol{\Sigma}_{k-1}^{(j)} \boldsymbol{\Omega}_k^H \boldsymbol{\Xi}_k^{(j)H} \left(\mathbf{R}_k^{(j)}\right)^{-1} \boldsymbol{\Xi}_k^{(j)} \boldsymbol{\Omega}_k \boldsymbol{\Sigma}_{k-1}^{(j)} \tag{5.46}$$

At each recursion *k*, the dominant computation in this particle filtering algorithm involves the $m \times A^P$ one-step Kalman filter updates for $\left(\mathbf{h}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)}\right)$. Since the *m* samplers operate independently and in parallel, the SMC detector is well suited for parallel implementations.

The residual resampling algorithm, which forms a new set of weighted samples $\left\{\left(\tilde{\mathbf{S}}_k^{(j)}, \tilde{k}_k^{(j)}, \tilde{w}_k^{(j)}\right)\right\}_{j=1}^m$ from original set $\left\{\left(\mathbf{S}_k^{(j)}, k_k^{(j)}, w_k^{(j)}\right)\right\}_{j=1}^m$ according to the algorithm 3.2.

## 5.3 Simulation Results

For the simulation of blind detection in MIMO-OFDM systems, the fading coefficients are assumed to be uncorrelated. All $L$ taps of each fading channel are Rayleigh distributed and normalized such that total energy is equal to unity.

For the simulation of blind detection in MIMO-OFDM system by particle filtering algorithm in MATLAB environment, the following parameters are used:

- Number of particles m= 50
- Modulation schemes: BPSK
- Number of transmitting antennas $P$=2
- Number of receiving antennas $Q$=2
- Number of subcarriers $N$=64
- Number of taps of each fading channel $L$=3
- Number of transmitted symbols=100000
- Number of Monte Carlo simulations $N_m$=100
- Resampling is done at every $k_0 = 5$ recursions.

Steps carried out for simulation of particle filtering algorithm for MIMO-OFDM system are:

1) Generate BPSK signals from alphabet set $A = \{-1, 1\}$ randomly and differentially encode them before transmitting from each antenna using equation (5.1).
2) Generate the fading coefficients and observations according to state space model of MIMO-OFDM system given by equations (5.13) & (5.14).
3) Generate the sequential Monte Carlo samples of transmitted symbols $\left\{S_k^{(j)}\right\}_{j=1}^{m}$ with corresponding importance weights $\left\{w_k^{(j)}\right\}_{j=1}^{m}$ at $k^{th}$ recursion by using the algorithm 5.1.
4) Do resampling by using algorithm (3.2) whenever $k$ is a multiple of $k_0$ else go back to step 3.

5) For each $\mathbf{a}_i \in A^P$, the a posteriori symbol probability $p\left(\mathbf{d}_k = \mathbf{a}_i / \gamma_k\right)$ is calculated using the equation (5.42).

6) The symbol is decoded by maximization of a posterior probability $p\left(\mathbf{d}_k = \mathbf{a}_i / \gamma_k\right)$ and bit error rate (BER) is calculated between transmitted symbols and decoded symbols.

Steps from 1 to 6 are repeated for each independent Monte Carlo run and BER is averaged over all Monte Carlo runs.

Fig. 5.2 shows BER performance of the particle filtering in blind detection of MIMO-OFDM system and additive Gaussian noise with BPSK modulation. Besides this, the performance of MLSE receiver with perfect channel state information, which serves as a lower bound on the achievable performance for any blind receiver is also plotted. As SNR is varied from 0dB-25dB, the BER of particle filtering decreases from 0.2143 to 0.0003 while BER of known channel bound decreases from 0.1974 to 0.00015. It may be seen that there is a close similarity between the known channel bound and the particle filtering method. For instance at SNR of 15dB, particle filtering method gives BER of 0.0038 while known channel bound gives BER of 0.0026.

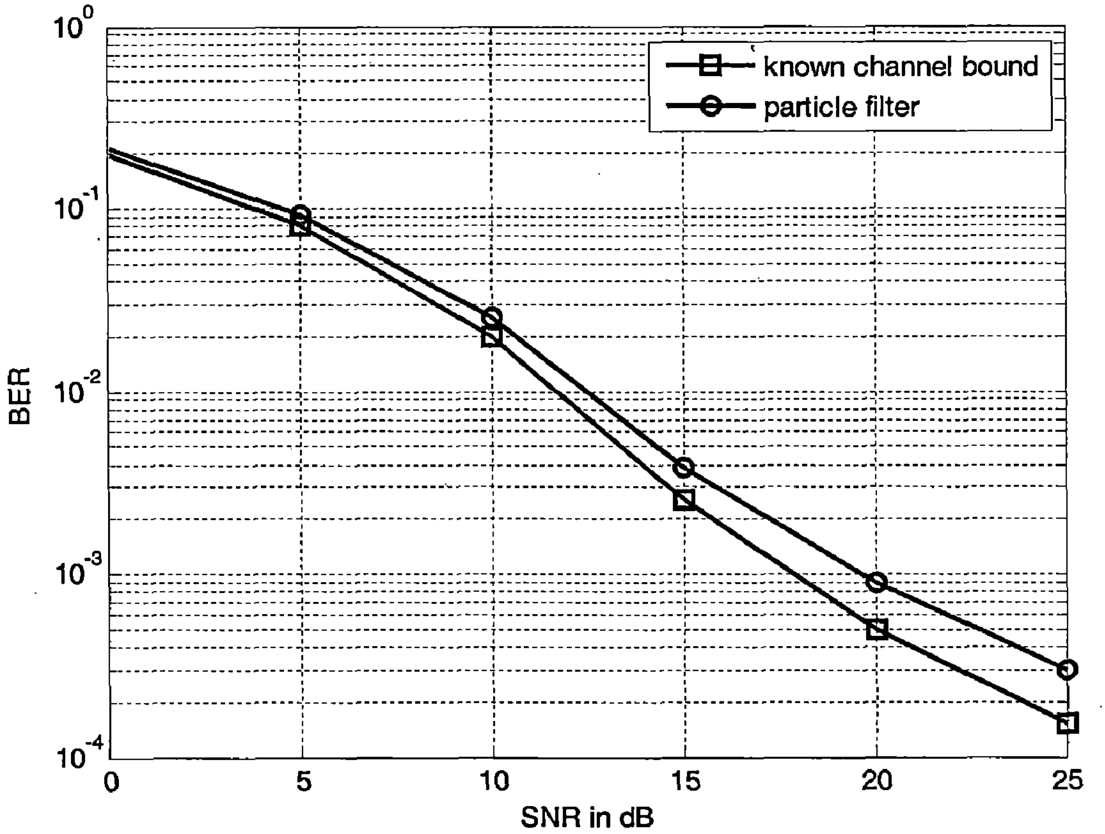



**Figure 5.2** BER performance of MIMO-OFDM system

# Chapter 6
# CONCLUSIONS

Optimal estimation for non-linear non-Gaussian state-space models does not typically admit analytic solution. Since their contribution in 1993, particle filtering methods have become a very popular class of algorithms to solve these estimation problems numerically in an online manner, i.e. recursively as observations become available, and are now routinely used in various 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 information data based on the received signals in the presence of various unknown channel distortions. The optimal solutions to these problems are often too computationally complex to implement by conventional signal processing methods. The recently emerged particle filtering methods are extremely powerful numerical techniques for tackling wireless signal processing problems. Particle filters are sequential Monte Carlo methods which can be applied to any state space model and generalizes the Kalman filtering methods. Particle filter uses the concept of sequential importance sampling (SIS) for the recursive computation of a posteriori pdf by drawing of samples from the importance density with corresponding importance weights. This dissertation work is aimed at the application of particle filtering for blind detection in SISO, MIMO, OFDM and MIMO-OFDM systems. The conclusions drawn based on the simulation results are as follows:

*Particle Filtering for Blind Detection in SISO and MIMO Systems*

We have used the state space model approach for deriving the particle filtering algorithm for the blind detection in SISO and MIMO systems with the use of Kalman filtering algorithm. As simulation results of SISO system 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 12 dB for zero delay particle filtering. It is seen that the delayed weighted scheme shows improvement in performance of 5 dB at BER of $10^{-3}$ when compared to that of particle filtering and is close to the known channel bound. The BER performance of MIMO system over unknown fading channels obtained through simulation is within 1 dB of the known channel bound (MLSE). The particle filtering with zero delay shows 6dB improvement in performance for MIMO system when compared to SISO system at BER of $10^{-2}$.

*Particle Filtering for Blind Detection in OFDM Systems*

We have used the state space model approach for deriving the particle filtering algorithm for the blind detection OFDM systems over unknown frequency-selective fading channels. As the simulation show, the BER and WER performance of the delayed weight methods is close to the known channel bound. Typically, the performance of delayed weight method is within 1 dB to the known channel bound for BER $10^{-3}$. It is also seen that the delayed weight method gives 4 dB improvement in performance for BPSK modulation when compared to QPSK modulation for OFDM systems for BER of $10^{-2}$.

*Particle Filtering for Blind Detection in MIMO-OFDM Systems*

The particle filtering approach for blind detection in MIMO-OFDM systems is exploited in this dissertation. From the simulation results, it may be seen that there is a close similarity between the known channel bound and the zero delay particle filtering method. The performance of particle filtering with zero delay is within 1.5 dB to the known channel bound for BER of $10^{-3}$. It is also evident that there is a 4 dB improvement in performance for MIMO-OFDM system when compared to that of OFDM system for BER of $10^{-3}$.

*Future work*

Monte Carlo methods mainly 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. A study on different MCMC methods and its applications to wireless communications is a topic of significant interest. Sequential Monte Carlo approach is powerful in statistical signal processing but its complexity is usually very high. Thus the low-complexity SMC algorithms for blind detection in wireless communications is a possible line of future work. Target tracking by particle filtering in binary sensor networks is also a topic of significant interest.

# REFERENCES


[1] Zhe Chen, "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond" Adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada, 2003. Available: http://soma.crl.mcmaster.ca/~zhechen/download/ieee_bayesian.ps.

[2] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim 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] Athanasios Papoulis, S.Unnikrishna Pillai, "Probability, Random Variables and Stochastic Processes", 4th edition, McGraw-Hill, 2002.

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

[5] Simon Haykin, "Adaptive Filter Theory", 4th edition, Pearson Education, 2002.

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

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

[9] Simon Haykin, Kris Huber, and Zhe Chen, "Bayesian Sequential State Estimation for MIMO Wireless Communications," Proceedings of the IEEE, vol. 92, no.3, March, 2004.

[10] Petar M. Djuric, Jayesh H. Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Monica F. Bugallo, and Joaquin Miguez, "Particle Filtering" IEEE Signal Processing Magazine, vol.20, pp. 19-38, September, 2003.

[11] Ying Liu, Benping Wang, Wang He, Jing Zhao, Zhi Ding, "Fundamental Principles and Applications of Particle Filters," Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, vol. 2, pp.5327-5331, 21-23 June, 2006.

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

[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.G. Proakis, "Digital Communications," 4th edition, McGraw-Hill, 2003.

[15] Theodore S.Rappaport, "Wireless Communications," Second edition, Eastern Economy Edition, 2006.

[16] Rong Chen, Xiaodong Wang and Jun 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.

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

[18] Arogyaswami Paulraj, Rohit Nabar and Dhananjay Gore, "Introduction to Space-Time Wireless Communications," 1st edition, Cambridge University Press, 2003.

[19] Dong Guo and Xiaodong Wang, "Blind Detection in MIMO Systems via Sequential Monte Carlo," IEEE Journal on Selected areas in Communications, vol.21, no.3, April 2003.

[20] Shinsuke Hara, Ramjee Prasad, "Multi carrier techniques for 4G mobile communications," 2nd edition, Artech house publishers, 2003.

[21] K.Fazel and S.Kaiser, "Multi-Carrier and Spread Spectrum Systems," 1st edition, Wiley publishers, 2003.

[22] Zigang Yang and Xiaodong 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.

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

[24] Petar M. Djuric, Mahesh Vemula, Monica F. Bugallo, "Target Tracking by Particle Filtering in Binary Sensor Networks," IEEE Transactions on Signal Processing, vol.56, no.6, June, 2008.

[25] F.Gustafsson, F.Gunnarsson, N.Bergman, U.Forsell, J.Jasson, R.Karlson, and Per-Jphan Nordlund, "Particle Filters for Positioning, Navigation and Tracking," IEEE Transactions on Signal Processing, vol.50, no.2, February, 2002.

[26] B.Schon, "Estimation of Nonlinear Dynamic Systems – Theory and Applications," Dissertations No. 998, Elect. Eng. Dept., Linkoping Univ., Linkoping, Sweden, 2006.

[27] Bolic, M.; Djuric, P.M.; Sangjin Hong, "New Resampling Algorithms for Particle Filters," Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03), 2003 IEEE International Conference on vol.2, no., pp. II-589-92 vol.2, 6-10 April, 2003.

[28] Hol, Jeroen D.; Schon, Thomas B.; Gustafsson, Fredrik, "On Resampling Algorithms for Particle Filters," Nonlinear Statistical Signal Processing Workshop, 2006 IEEE, vol.1, no.1, pp.79-82, 13-15 September, 2006.

[29] G.Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Non-linear State Space Models," Journal of Computational and Graphical Statistics, vol.5, no.1, pp.1-25, 1996.

[30] R.Haeb and H.Meyr, "A Systematic Approach to Carrier Recovery and Detection of Digitally Phase Modulated Signals on Fading Channels, " IEEE Transactions on Communications, vol.3, no.7, pp.748-754, July, 1989.

[31] 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.

[32] David 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.

[33] Rong Chen and Jun S. Liu, "Mixture Kalman Filters," J.R. Statist, Soc.B, part-3, pp. 493-508, 2000.

[34] J.J. van de Beek, O. Edfords, M. Sandell, S.K. Wilson and P.O.Borjesson, " On Channel Estimation in OFDM Systems, " in Proceedings of IEEE Vehicular Technology Conference, vol.2, pp. 815-819, Chicago, September, 1995.

[35] Y. Li, J. Cimini, and N. Sollenberger, " Robust Channel Estimation for OFDM Systems with Rapid Dispersive Fading Channels," IEEE Transactions on Communications, vol.46, no.7, pp. 902-915, July, 1998.

[36] Jiang Yue, Kyeong Jin Kim, Tony Reid, Jerry D.Gibson, "Joint Semi-Blind Channel Estimation and Data Detection for MIMO-OFDM systems," IEEE 6th CAS Symp. on Engineering Technologies: Mobile and Wireless Comm., Shanghai , China, pp.709-712, May 31-June 2, 2004.

[37] Kyeong Jin Kim, Tony Reid, Ronald A. Iltis, "A Sequential Monte-Carlo Kalman Filter Based Delay and Channel Estimation Method in the MIMO-OFDM System," Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th , vol.1, pp. 573-577, 26-29 September, 2004.

[38] Kyeong Jin Kim, Jiang Yue, Ronald A. Iltis, Jerry D.Gibson," A QRD-M/Kalman Filter Based Detection and Channel Estimation Algorithm for MIMO-OFDM Systems, " IEEE Transactions on Wireless Communications, vol.4, no.2, pp.710-721, March, 2005.